



INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Low Delay Distributed Video Coding

António Miguel dos Santos Francês Tomé

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

Júri

Orientador: Prof. Fernando Pereira

Outubro 2009

Acknowledgments

I would like to thank all the people who supported me, and pressured me to go further and further.

A special thanks for Professor Fernando Pereira for the never ending patience, precious advice and his constant concern about the work being performed. Without his guidance and knowledge this Thesis would not have been possible.

My parents, for their endless support in every aspect, both in good and bad moments.

To all my friends, that tried to keep me in good mood and stress free, whenever possible.

Thank you all.

Abstract

Distributed Video Coding is a new branch for video coding solutions, build upon two fundamental theorems, the Slepian-Wolf and Wyner-Ziv theorem. With the development of these theorems a new paradigm appeared, one that would provide an answer to some applications requirements. Advantages like flexible allocation of complexity between encoder and decoder, improved error resilience and exploitation of multiview correlation without cameras inter communication, made DVC based codecs, good alternatives in emerging application such as wireless video surveillance. The first two practical approaches were created around 2002, by the Stanford University and University of California, Berkeley giving birth to the Stanford DVC codec and the PRISM respectively. The Stanford DVC architecture was later adopted and improved by many groups in order to develop more efficient codecs. An example of this development is the creation of the state-of-the-art DISCOVER DVC codec, as it shows good RD performance results when compared with other benchmarks, both predictive and DVC based. This codec, does not meet delay requirements though, as it uses interpolation techniques on top of the feedback channel, common to Stanford based architectures.

Considering all the above, this Thesis proposes to fill that gap, following two objectives i) creating an efficient and practical DVC codec, ii) low delay driven, based on the Stanford DVC approach. The results to be presented will demonstrate that the developed codec complies with the proposed objectives.

Keywords: Distributed Video Coding, Wyner-Ziv, low delay, motion extrapolation, side information refinement.

Resumo

Codificação Distribuída de Vídeo (CDV) é um novo ramo de soluções de codificação de vídeo, assente em dois teoremas fundamentais, o teorema de Slepian-Wolf e o teorema de Wyner-Ziv. Com o desenvolvimento destes teoremas surgiu um novo paradigma capaz de responder a certo tipo de requisitos provenientes de certos tipos de aplicações. Vantagens como alocação flexível de complexidade entre codificador e decodificador, robustez contra erros, exploração de correlação a nível de multiview, sem que as câmeras comuniquem entre si, tornaram os codecs CDV numa boa solução em aplicações como video vigilância wireless. Um exemplo dos desenvolvimentos nesta área é a criação do codec DISCOVER DVC, considerado state-of-the-art no espectro CDV. A sua RD performance demonstra a sua eficiência, pois apresenta bons resultados quando comparado com outros esquemas, quer preditivos, quer CDV. No entanto, devido a uso de técnicas de interpolação e uso de canal de feedback, tornam-no uma má escolha quando proposto para sistemas com requisitos de baixo atraso.

Nesse sentido, esta Tese tenta colmatar esta falha, ao apresentar dois objectivos i) criação de um codec CDV eficiente, ii) cumpra requisitos de baixo atraso. Os resultados mais tarde apresentados, demonstrarão o cumprimento desses mesmos objectivos.

Palavras Chave: Codificação Distribuída de Vídeo, Wyner-Ziv, baixo atraso, extrapolação de movimento, refinamento de informação adicional.

Table of Contents

Acknowledgments	i
Abstract	ii
Resumo	iii
Table of Contents.....	iv
Index of Figures	vii
Index of Tables.....	ix
List of Acronyms.....	x
Chapter 1 - Introduction.....	1
1.1. Context and Motivation	1
1.2. Distributed Video Coding Theoretical Foundations	2
1.3. Distributed Video Coding Practical Solutions	5
1.4. Thesis Objectives	6
1.5. Thesis Structure	6
Chapter 2 - Reviewing Distributed Video Coding	8
2.1. Classifying Distributed Video Coding Solutions.....	8
2.1.1. First Dimension: Monoview versus Multiview	9
2.1.2. Second Dimension: Low Delay versus High Delay	10
2.1.3. Third Dimension: Frame Based versus Block Based	10
2.1.4. Forth Dimension: Feedback Channel versus No Feedback Channel.....	11
2.2. Most Relevant Distributed Video Coding Solutions	11
2.2.1. The Transform Domain DISCOVER DVC Codec	12
2.2.1.1. Objectives.....	12
2.2.1.2. Approach and Architecture	12
2.2.1.3. Main Tools	14
2.2.1.4. Performance Evaluation	17

2.2.1.5. Summary	19
2.2.2. Low Delay Pixel Domain IST DVC Codec	19
2.2.2.1. Objectives	19
2.2.2.2. Approach and Architecture	20
2.2.2.3. Main Tools	20
2.2.2.4. Performance Evaluation	21
2.2.2.5. Summary	22
2.2.3. Pixel Domain Low Delay DVC using Iterative Refinement Side Information Generation.	22
2.2.3.1. Objectives	22
2.2.3.2. Approach and Architecture	23
2.2.3.3. Main Tools	23
2.2.3.4. Performance	25
2.2.3.5. Summary	25
2.2.4. Motion Compensated Extrapolation Techniques for Side Information Generation	26
2.2.4.1. Objectives	26
2.2.4.2. Approach and Architecture	26
2.2.4.3. Main Tools	26
2.2.4.4. Performance Evaluation	29
2.2.4.5. Summary	31
Chapter 3 - Advanced Low Delay IST DVC Codec	32
3.1. Codec Architecture	32
3.2. Extrapolation-based Side Information Creation	36
3.2.1. Motion Estimation Sub-Module	37
3.2.2. Motion Field Smoothing Sub-Module	37
3.2.3. Motion Projection Sub-Module	38
3.2.4. Overlapping and Uncovered Areas Treatment Sub-Module	38
3.3. Correlation Noise Modeling	40
3.3.1. Residual Frame Computation	41
3.3.2. RDCT Computation and RDCT Frame Generation	43
3.3.3. RDCT Band b Variance Computation	43
3.3.4. α Parameter Estimation at DCT Band b Level	43
3.3.5. RDCT (u,v) DCT Coefficient Distance Computation	43
3.3.6. α Parameter Estimation at DCT Coefficient (u,v) Level	44
3.4. Performance Evaluation	44
3.5. Results and Analysis	46
3.5.1. RD Performance for GOP Size 2	46
3.5.2. RD Performance for Longer GOP Sizes	49

3.5.3. WZ Frames RD Performance.....	54
3.6. Final Remarks	55
Chapter 4 - Advanced Low Delay IST DVC Codec with Side Information Refinement.....	56
4.1. Codec Architecture	56
4.2. Side Information Refinement Algorithm	58
4.2.1. Initial DCT Domain Side Information Creation	58
4.2.2. Block Selection for Refinement	58
4.2.3. Candidate Blocks Searching	59
4.2.4. New Side Information Creation	60
4.3. Reconstruction Algorithm.....	61
4.3.1. Quantized DCT Bands	61
4.3.2. Unquantized DCT Bands	62
4.4. Performance Evaluation.....	63
4.4.1. RD Performance for GOP Size 2	63
4.4.2. RD Performance for Longer GOP Sizes	67
4.5. Final Remarks	70
Chapter 5 - Conclusions and Future Work.....	71
References.....	74

Index of Figures

Figure 1.1 – Traditional coding architecture with joint encoding and joint decoding.	2
Figure 1.2 – Slepian-Wolf scenario: two source sequences, X and Y, independently encoded but jointly decoded.	3
Figure 1.3 – Rate boundaries by the Slepian-Wolf theorem for independent encoding and joint decoding.	3
Figure 1.4 – Wyner-Ziv scenario with lossy coding	4
Figure 2.1 – Proposed classification tree for DVC solutions.	9
Figure 2.2 – DISCOVER DVC video codec architecture [20].	12
Figure 2.3 – Frame Interpolation Framework architecture [20].	16
Figure 2.4 – Sample frames for test sequences: a) Foreman (Frame 80); b) Hall Monitor (Frame 75); c) Coast Guard (Frame 60); d) Soccer (Frame 8).	18
Figure 2.5 – RD performance comparison for various test sequences (QCIF, 15 Hz, GOP=2) [20].	19
Figure 2.6 – Low Delay IST DVC codec architecture with side information extrapolation [24].	20
Figure 2.7 – Side information creation by extrapolation architecture [24].	21
Figure 2.8 – Motion projection assuming linear motion [24].	21
Figure 2.9 – RD Performance comparison between several GOP sizes for a) Foreman sequence, b) Galleon sequence [24]. ..	22
Figure 2.10 – DVC architecture for the iterative refinement technique for side information generation [25].	23
Figure 2.11 – Proposed iterative refinement side information generation architecture [25].	24
Figure 2.12 – PSNR comparison between several codecs [25].	25
Figure 2.13 – Building the candidate set: C is the current block while S and T represent the spatial and temporal candidates [34].	26
Figure 2.14 – Extrapolation scheme for forward motion estimation [21].	28
Figure 2.15 – Extrapolation scheme for backward motion estimation [21]	29
Figure 2.16 – Side information PSNR comparison for the a) Foreman and b) Coast Guard sequences: 1) SVFME; 2) SBME; 3) IBME; 4) BA.	31
Figure 3.1 – ALD-DVC architecture.	33
Figure 3.2 – Encoding/decoding orders for interpolation and extrapolation-based DVC schemes for GOP=4.	34
Figure 3.3 – Side Information Creation module.	36

Figure 3.4 – a) Side information after motion projection without motion vector field smoothening; b) Side information after motion projection with motion vector field smoothening; c) Side information after motion projection with motion vector field smoothening and treatment of overlapping zones and holes.	39
Figure 3.5 – Correlation noise modeling architecture.	40
Figure 3.6 – Motion projection process.	41
Figure 3.7 – Eight quantization matrices corresponding to the tested RD points.	45
Figure 3.8 – RD performance comparison for the Coastguard sequence using GOP size 2.	46
Figure 3.9 – RD performance comparison for the Foreman sequence using GOP size 2.	47
Figure 3.10 – RD performance comparison for the Hall Monitor sequence using GOP size 2.	47
Figure 3.11 – d RD performance comparison for the Soccer sequence using GOP size 2.	48
Figure 3.12 – ALD-DVC RD performance comparison for the Coastguard sequence, using GOP size 2, 4 and 8.	50
Figure 3.13 – ALD-DVC RD performance comparison for the Foreman, sequence using GOP size 2, 4 and 8.	50
Figure 3.14 – ALD-DVC RD performance comparison for the Hall Monitor sequence, using GOP size 2, 4 and 8.	51
Figure 3.15 – ALD-DVC RD performance comparison for the Soccer sequence, using GOP size 2, 4 and 8.	51
Figure 3.16 – PSNR versus Frame Number for Foreman sequence 15 Hz using GOP size 2 and $Q_i=4$	52
Figure 3.17 – PSNR versus Frame Number for Foreman sequence 15 Hz using GOP size 4 and $Q_i=4$	52
Figure 3.18 – PSNR versus Frame Number for Foreman sequence 15 Hz using GOP size 8 and $Q_i=4$	53
Figure 3.19 – PSNR versus Frame Number for Foreman sequence 15 Hz, using GOP size 2, 4, 8 and $Q_i=4$	53
Figure 3.20 – RD performance comparison for the Foreman sequence.	54
Figure 4.1 – The ALD-DVC SIR codec architecture.	57
Figure 4.2 – RD performance comparison for the Coastguard sequence using GOP size 2.	63
Figure 4.3 – RD performance comparison for the Foreman sequence using GOP size 2.	64
Figure 4.4 – RD performance comparison for the Hall Monitor sequence using GOP size 2.	64
Figure 4.5 – RD performance comparison for the Soccer sequence using GOP size 2.	65
Figure 4.6 – RD performance comparison for the Coastguard sequence using GOP size 2, 4 and 8.	67
Figure 4.7 – RD performance comparison for the Foreman sequence using GOP size 2, 4 and 8.	68
Figure 4.8 – RD performance comparison for the Hall Monitor sequence using GOP size 2, 4 and 8.	68
Figure 4.9 – RD performance comparison for the Soccer sequence using GOP size 2, 4 and 8.	69
Figure 4.10 – RD performance comparison for Hall Monitor sequence using GOP size 8.	70

Index of Tables

Table 2.1 – DISCOVER test conditions.....	18
Table 2.2 – PSNR comparison between different side information creation schemes.....	30
Table 3.1 - Performance results for the ALD-DVC codec.....	46
Table 4.1 – RD performance results for ALD-DVC SIR codec.....	63

List of Acronyms

3DRS	3-D Recursive Search
AC	Alternate Current
ALD-DVC	Advanced Low Delay Distributed Video Coding
AVC	Advanced Video Coding
BER	Bit Error Rate
BA	Bilateral Filtering
CARS	Content Adaptive Recursive Search
CNM	Correlation Noise Modeling
CRC	Cyclic Redundancy Check
dB	Decibel
DC	Direct Current
DCT	Discrete Cosine Transform
DISCOVER	DIStributed COding for Vídeo sERVICES
DSC	Distributed Source Coding
DVC	Distributed Video Coding
GOP	Group Of Pictures
IBME	Improved Backward Motion Estimation
IDCT	Inverse Discrete Cosine Transform
IEC	International Electrotechnical Commission
ISI GEN	Initial Side Information GENeration
ISO	International Organization for Standards
IST	Instituto Superior Técnico
ITU-T	International Telecommunications Union - Telecommunications standardization
FSBM	Full Search Block Matching
FVFME	First Variation for Forward Motion Estimation
LDPC	Low-Density Parity-Check
LDPCA	Low-Density Parity-Check Accumulate
LLR	Logarithmic Likelihood Ratio
MAD	Mean Absolute matching Difference
ME	Motion Estimation
MSE	Mean Squared Error
MPEG	Motion Picture Experts Group

MVC	Multiview Video Coding
MX	Motion Extrapolation
PRISM	Power-efficient, Robust, hIgh-compression, Syndrome-based Multimedia coding
PSNR	Peak Signal Noise Ratio
QCIF	Quarter Common Intermediate Format for images (144 lines by 176 columns)
RD	Rate Distortion
SBME	Simple Backward Motion Estimation
SI	Side Information
SIR	Side Information Refinement
SPA	Sum Product Algorithm
SVFME	Second Variation for Forward Motion Estimation
SW	Slepian-Wolf
VQEG	Video Quality Experts Group
WMAD	Weighted Mean Absolute matching Difference
WZ	Wyner-Ziv

Chapter 1

Introduction

1.1. Context and Motivation

Video coding associated with audio coding have been playing an important role in the realm of multimedia services. This is due to as the increased usage of audiovisual content and the growing deployment of devices such as digital TVs, mobile phones, surveillance cameras and Internet applications requiring more efficient codecs to cope with the ever growing quality needs spending as less rate as possible, this means maximizing the rate-distortion (RD) performance. The main tools used by the most common video coding standards, notably the ITU-T H.26x and ISO/IEC MPEG-x families of standards largely used around the world, are the following:

- Motion compensated temporal prediction between video frames to exploit as best as possible the temporal redundancy; if a video sequence is very stable or with low motion, the temporal redundancy is high, and compression efficiency increases.
- Spatial redundancy exploitation using transform coding, namely the Discrete Cosine Transform (DCT).
- Quantization of the transform (typically DCT) coefficients to effectively reduce the video information which is not perceived by the human visual system; for example, the high frequency components are not as relevant as the low frequency components and, thus, are typically quantized with a larger quantization step reducing the associated rate.
- Entropy coding to exploit the statistical redundancy in the created symbols.

The use of the largely available video coding standards typically implies a rather high level of computational complexity in the (non-normative) encoder where all the redundancy and irrelevancy exploitation processes are performed; the generated symbols are sent to a normative decoder, usually much less complex. This high encoding complexity versus low decoding complexity coding paradigm, typical of these so-called *predictive coding schemes*, is particularly adequate for one-to-many topology scenarios, like broadcasting in whatever transmission channel, where there is a centralized generated encoded video stream sent to thousands of (decoding) users. This video stream is typically encoded by one of the available video coding standards mentioned above and transmitted through one of a growing variety of channels, and decoded multiple times,

depending on the number of users consuming it. Thus, in a world controlled by the need for high profits and low costs, having a coding paradigm characterized by a complex encoder with high cost and multiple decoders with low complexity and consequently low cost, fits this type of application scenarios very well. But what about a situation where there are many encoders that need to send streams to a central decoder like in the video surveillance type of scenario ? Imagine, for example, an application scenario with multiple wireless cameras performing the surveillance of a certain location. Following the predictive coding paradigm with high complexity encoders and low complexity decoders, each and every one of these cameras would be rather expensive, depending on the level of processing performed. This cost would go even higher if this system would try to exploit the redundancy between the various camera views, implying inter-camera communication and a centralized encoding process. Other emerging examples regard wireless video cameras and low power video sensor networks that under the traditional video paradigm would prove to be a poor investment considering all the costs implied. These application examples highlight application requirements which do not seem well addressed by the available predictive video coding paradigm.

However, around 2002, some research groups studied the possibility to implement in practice video codecs based on the Slepian-Wolf [3] and Wyner-Ziv theorems [4] which basically open the doors to the possibility to develop source codecs with other types of complexity allocations without any RD performance penalty. This new coding paradigm applied to video information led to a new research area, well known as Distributed Video Coding (DVC). Thus, a new coding paradigm emerged eventually providing solutions that predictive schemes could not provide, notably shifting the complexity from the encoder to the decoder, and allowing a more flexible complexity allocation, theoretically with the same compression efficiency. In this way, there is the possibility that systems like video surveillance among others, requiring low complexity encoders, robustness to packet losses, high compression efficiency and, sometimes, even low delay, could be implemented under the novel DVC paradigm, greatly reducing costs. Also inter-view redundancy exploitation may be performed using a DVC approach without the need for inter-camera communications, thus offering low encoding complexity with the heaviest processing being performed at the decoder. Thus, the emergence of this video coding approach provides coding alternatives for applications that could not get an efficient solution with the traditional video coding solutions.

1.2. Distributed Video Coding Theoretical Foundations

Assuming traditional coding schemes, if two statistically dependent sequences X and Y are jointly encoded and jointly decoded, it is possible to obtain a perfect reconstruction of X and Y , if the total rate, R , spent is equal to the joint entropy of these sequences, $H(X,Y)$, as shown in Figure 1.1. If the encoding is not done using the usual joint coding process, this means if the redundancy is not exploited at the encoder but rather the two sequences are independently encoded to simplify the encoding process, what is the rate required to perfectly reconstruct the sequences at the decoder ? Is there any penalty regarding the joint entropy rate characterizing the joint encoding, joint decoding case ?

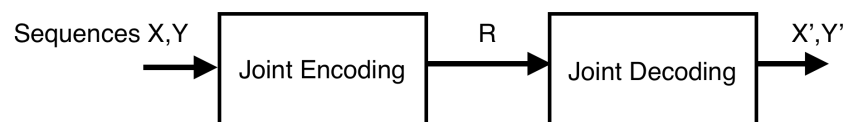


Figure 1.1 – Traditional coding architecture with joint encoding and joint decoding.

The answer for the question above is in the Slepian-Wolf theorem developed in the seventies [3]. The main idea behind this theorem is that given two statistically dependent, discrete random sequences, X and Y , independently and identically distributed (i.i.d.), it is possible to jointly decode them at the decoder, even though they were not jointly encoded, using a rate R equal to or greater than the joint entropy of those same sequences. An illustration of the coding scenario addressed by the Slepian-Wolf theorem is presented in Figure 1.2.

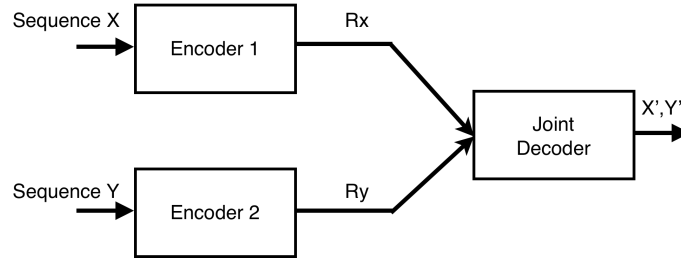


Figure 1.2 – Slepian-Wolf scenario: two source sequences, X and Y , independently encoded but jointly decoded.

According to the Slepian-Wolf theorem, the rate bounds for a vanishing error probability considering two sequences X and Y coded as in Figure 1.2 is given by:

$$R_x \geq H(X/Y) \quad (1.1)$$

$$R_y \geq H(Y/X) \quad (1.2)$$

$$(R_x + R_y) \geq H(X, Y) \quad (1.3)$$

Note that equations (1.1), (1.2) and (1.3) are represented in Figure 1.3 in a chart form allowing a more graphical interpretation of the Slepian-Wolf theorem. In this figure, two zones can be distinguished:

- **Zone 1** – By performing independent encoding/decoding, it is possible to reconstruct the sequences X and Y with no errors, even though the rate spent is higher than necessary.
- **Zone 2** – By performing independent encoding but joint decoding, it is possible to reconstruct the sequences X and Y with a vanishing error probability; the advantage is that the rate spent in this zone is lower than the rate in Zone 1.

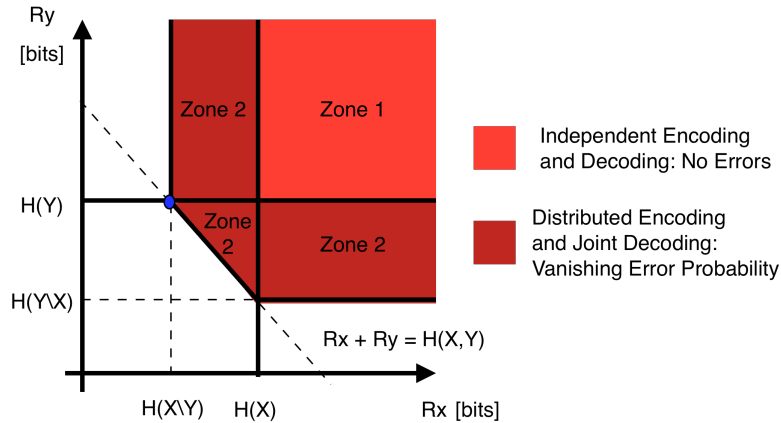


Figure 1.3 – Rate boundaries by the Slepian-Wolf theorem for independent encoding and joint decoding.

Hence, by using the minimum total coding rate defined in equation (1.3), i.e. the same as for joint encoding and joint decoding, it is possible to jointly decode the two correlated sequences not jointly encoded, achieving at the same time a vanishing error probability. In practice, this means it is possible to (almost) perfectly reconstruct the two sequences, X and Y , at the decoder using what is usually called *Slepian-Wolf coding*; this is typically called, a distributed coding ‘lossless’ case, in practice neglecting the small error probability. This type of coding scenario and the associated theorem are known as distributed source coding (DSC) since the source is coded in a distributed way as it is divided in correlated ‘components’, with their correlation not being exploited at the encoder. When the information to be coded is video, this approach is well known as distributed video coding (DVC).

The Slepian-Wolf coding scenario presented is deeply related to channel coding. Considering two correlated sequences, X and Y , where Y is understood as an erroneous or corrupted version of X , it is possible to encode X by correcting the errors in Y using channel coding techniques, i.e. by encoding X independently and jointly decoding it with the help of sequence Y , taken as side information, it is possible to obtain a (almost) perfect reconstructed version of X .

Later, in 1976, A. Wyner and J. Ziv studied a particular case of distributed source coding, related to a particular point in Figure 1.3, the blue dot point where the rates are $H(X/Y)$ and $H(Y)$ for R_x and R_y , respectively, originating the Wyner-Ziv theorem [4]. This particular point/situation implies that sequence Y , correlated to X , is independently encoded and decoded while the sequence X is independently encoded but jointly decoded using Y as auxiliary side information available at the decoder, as shown in Figure 1.4. It is important to note the Wyner and Ziv study refers to the lossy coding of sequence X which corresponds to a more practical and relevant situation in the context of video coding considering the real world needs.

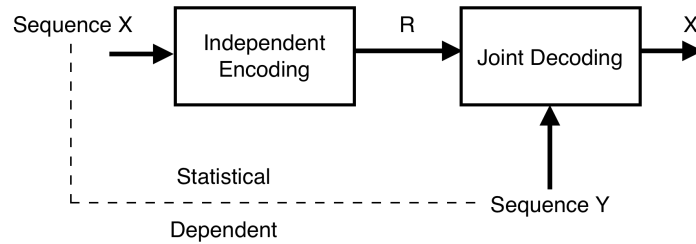


Figure 1.4 – Wyner-Ziv scenario with lossy coding

The Wyner-Ziv theorem states that when performing independent encoding of sequence X with side information Y available at the decoder, as shown in Figure 1.4, there is no coding efficiency loss, regarding the traditional coding schemes, i.e. using joint encoding/decoding, even if the coding process is lossy. For this to hold, it assumes that the X and Y sequences are jointly Gaussian and memoryless and a Mean-Squared Error (MSE) distortion measure is used [4].

As described above, DVC is a particular case of DSC when taking a video sequence as the source data. Thus, considering the Slepian-Wolf and Wyner-Ziv theorems in a video coding context, it can be easily concluded that a new variety of application scenarios can be addressed with this new video coding paradigm. Theoretically, it is possible to take two correlated video sources, whatever they are, encode them independently and jointly decode them while approaching the

coding efficiency of conventional predictive coding schemes, this means the joint entropy. It is important to note that this works in theory, but the theorems do not indicate the practical ways to achieve this level of compression efficiency leaving this task to video coding researchers. One more note regarding the side information Y available at the decoder when performing video coding is that even though it was never told how it is compressed, any type of traditional video coding solution may be used, notably the MPEG-x and H.26x standards.

This technology is still very recent and there is often a tendency to assume unpractical scenarios. For example, the theory assumes that the correlation between X and Y is known at the encoder, bringing higher compression efficiency, but this is rather unpractical as it implies a high level of encoding complexity, defeating a main purpose of the DVC paradigm: low encoding complexity.

Nevertheless, DVC architectures bring benefits to video coding applications, as the independent encoding of a sequence X generally results in the absence of temporal prediction loop, lowering the complexity of the encoder and avoiding the temporal error propagation effect, typical of predictive coding. Other benefits brought by the DVC approach are:

- The flexible allocation of global video codec complexity, making the encoder more or less complex and consequently the decoder less or more complex, respectively.
- Improved error resilience due to the absence of the prediction loop, as errors cannot propagate from one frame to another in its absence. Naturally, there are pixels being used from previous decoded frames at given times but, as will be detailed in the next chapters, eventual errors are corrected by the DVC approach, regardless of the quality used.
- Codec independent scalability since upper layers do not have to rely on precise lower layers.
- Exploitation of multiview correlation at the decoder, without encoders communicating among them, thus lowering the encoder complexity. This multiview correlation exploitation, if done at the encoder, would imply more complex processing tools, increasing the complexity of the encoder, as also cameras would have to communicate among them. At the light of the DVC paradigm, this is not needed as it is the decoder who is in charge of exploiting the inter-view correlation, thus maintaining low encoding complexity.

1.3. Distributed Video Coding Practical Solutions

Around 2002, advances regarding practical DVC codecs have emerged, giving birth to the two well-known approaches in this field, namely the Stanford [1], [5], [6] and Berkeley [7] DVC video coding solutions.

The Stanford DVC video coding solution, developed at the Stanford University, is characterized by a frame based architecture, turbo codes and a feedback channel used to control the rate at the decoder while the Berkeley DVC solution, also known as PRISM (Power-efficient, Robust, hIgh-compression, Syndrome based Multimedia coding), is block based driven with decoder motion estimation. Although both architectures appeared almost simultaneously, the Stanford architecture proved to be more interesting when developing new tools to increase RD performance. Consequently, this is the most common architecture used as start point to develop other DVC solutions, using different tools depending on the application requirements, but always aiming for high RD performance gains. Over the years, techniques such as transform domain DCT (and not anymore pixel domain), use of LDPC (Low-Density Parity-Check) codes in the Slepian-Wolf codec (and not the usual turbo codes), better reconstruction algorithms and more efficient key frames coding, spending less rate, notably by

using state-of-the-art coding solutions like H.264/AVC Intra, have been developed/perfected to increase the efficiency of Stanford based DVC codecs.

This DVC approach is also capable of addressing the needs of applications with low latency/delay needs extrapolation based side information creation techniques and not interpolation based side information creation which generate the sequence Y using past and future previously decoded frames. More precisely:

- **Interpolation based DVC codecs** – Interpolation implies delay caused by the wait for future frames, needed to determine the side information sequence Y . This delay is compensated by the fact that, in principle, the motion estimation process becomes more reliable, leading to a better interpolated frame, which brings gains in terms of RD performance when compared to extrapolation based DVC based codecs.
- **Extrapolation DVC Based Codecs** – Delay is not introduced when using extrapolation techniques as they generate Y using only past frames. A trade-off is to be expected, as the motion estimation only using past frames is not as accurate as using past and future frames, as in the interpolation techniques. This usually means RD performance losses for the extrapolation DVC based codecs when compared with the interpolation based ones.

More detailed information regarding the Stanford architecture is provided in the next chapter, as this is the core of the codec developed by the author of this Thesis.

1.4. Thesis Objectives

In the context previously described, the objective of this Thesis is to develop a novel DVC solution based on the Stanford architecture mentioned above [1], with two main requirements:

- **High Efficiency** – The new DVC codec shall perform as best as possible in terms of compression efficiency, notably when compared with existing DVC solutions and standard based video coding solutions for a similar encoding complexity.
- **Low Delay** – The new DVC codec shall present a low algorithmic delay, which typically means that the decoding of any frame should not depend on any future frames. Note that state-of-the-art DVC codecs usually use interpolation based side information creation schemes in order to obtain better RD performances, such as the DISCOVER DVC codec [12]. The need for future frames to more accurately interpolate the side information frames adds delay; hence, a low delay driven DVC codec has to resort to other tools such as extrapolation techniques in order to avoid this delay.

In this context, the author has designed, implemented and evaluated under relevant test conditions a novel efficient, low delay DVC solution based on the Stanford architecture. To achieve the objectives, the DVC codec developed in this Thesis uses as its core the IST DVC Interpolation codec, an evolution of the DISCOVER DVC codec, both based on the Stanford early DVC approach. This core regards the coding architecture, the coding tools and also the corresponding software.

1.5. Thesis Structure

To efficiently present the work developed by the author, this Thesis is divided into five main chapters:

- Chapter 1 has the very important purpose of defining the context and motivation that led to this Thesis. It also gives basic insights on the Distributed Video Coding theoretical foundations by introducing the two most important

theorems, the Slepian-Wolf and Wyner-Ziv theorems, and on the first practical DVC solutions. Finally, Chapter 1 defines the objectives and structure for this Thesis.

- Chapter 2 proposes a classification taxonomy for available DVC solutions using some relevant dimensions, notably the number of views, the delay, the spatial coding support and the availability or not of a feedback channel. It also provides a detailed review on relevant DVC solutions, such as the DISCOVER DVC codec and other extrapolation based DVC codec solutions considered more interesting due to the adopted low delay requirement.
- In Chapter 3, a description of the Advanced Low Delay DVC codec developed by the author of this Thesis is presented in detail, notably the codec architecture, main tools and performance results. This chapter concentrates on two major topics and associated tools, crucial to the practicality of the codec: the side information extrapolation techniques and the correlation noise model, directly associated with the high efficiency and low delay requirements defined above.
- Chapter 4 presents an evolution of the Advanced Low Delay DVC codec introduced in Chapter 3, as this codec is improved by integrating a new set of tools that allow boosting the RD performance based on the idea of refining the side information along the decoding process. Thus, this chapter provides detailed information on the novel solution, notably the improved codec architecture, and the main tools, and studies after its performance.
- In Chapter 5, final remarks and conclusions are presented; also possible future developments for the DVC codecs developed by the author of this Thesis are proposed.

Chapter 2

Reviewing Distributed Video Coding

In recent years, distributed video coding (DVC) has been emerging as an alternative video coding approach to the traditional and largely deployed predictive video codecs, as adopted by most MPEG and ITU-T standards. As an emerging video coding technology, the main DVC objective at this stage is to ‘close the gap’ in terms of rate–distortion (RD) performance regarding predictive coding as the theoretical potential of DVC systems has not been fully tapped and, thus, there are large margins of improvement for the currently available DVC solutions.

The main objective of this chapter is to provide a short overview on the current DVC landscape with especial emphasis on low delay DVC solutions since they are the target of this Thesis. With this objective in mind, a classification tree for DVC solutions is first proposed; after, some relevant DVC solutions are reviewed, setting the ground for the choices to be made in the chapters to follow.

2.1. Classifying Distributed Video Coding Solutions

When analyzing the DVC landscape, many solutions are already available following different architectural approaches or focusing on different functionalities. As for most technical domains, the various DVC solutions can be clustered and classified depending on the main technical approach, concepts and tools used. Although this type of classification is not unique, having some classification tree helps interested people to better understand the relations between the various solutions and to get a more complete knowledge of the DVC solutions landscape.

As shown in Figure 1, the classification tree proposed in this chapter to structure and organize the DVC solutions is based on four classification dimensions, which are associated to the following concepts:

- **Number of camera views** – This dimension regards the number of cameras views to code, notably if the system is monoview or multiview.
- **Delay** – This dimension regards the coding delay, notably if the solution fulfils or not low delay requirements.

- **Basic spatial coding support** - This dimension regards the type of basic spatial coding support, notably frame or block based.
- **Feedback channel** - This dimension regards the exploitation or not of a feedback channel which has implications in terms of real-time performance.

Although there are other classification trees that may be equally adequate or more focused on other classification dimensions, what is most important here is to provide a way to organize the DVC solutions landscape in order their main commonalities and differences become more evident. Since the classification tree is similar for monoview and multiview systems, Figure 2.1 only shows in full detail the classification tree for the monoview side; the multiview branch follows exactly the same structure, this means the same levels in the same order.

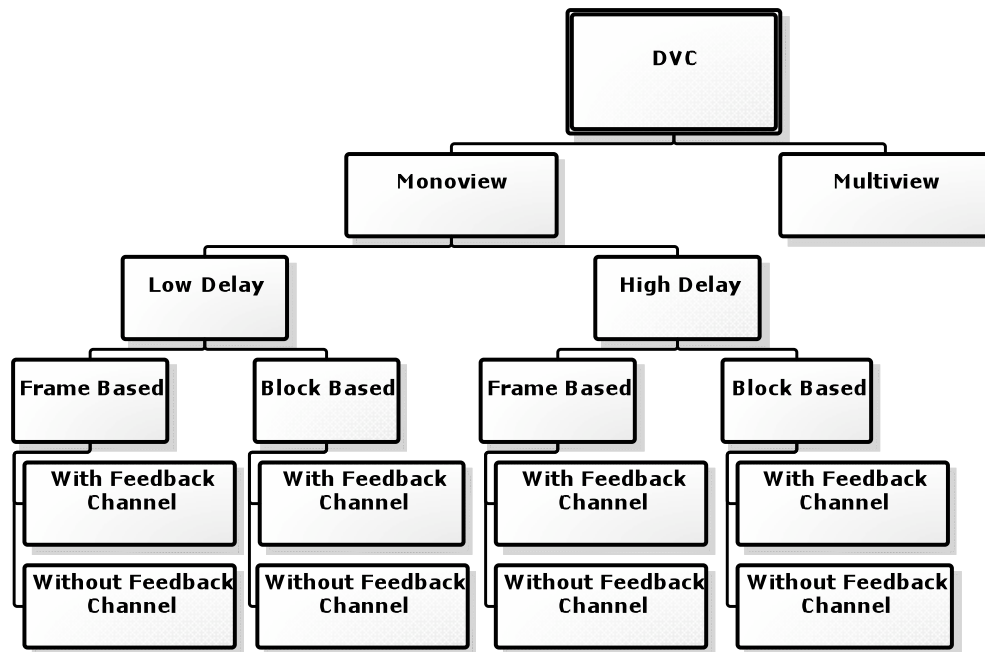


Figure 2.1 – Proposed classification tree for DVC solutions.

2.1.1. First Dimension: Monoview versus Multiview

Monoview systems are the most common as they use only one source of video information, originated by a single camera, for example, a cell phone or a security camera. Monoview DVC solutions create a single bitstream sent to the decoder, where the entire heavy processing (decoding) is to be performed. All the solutions under review in this chapter are monoview systems since this is the type of system addressed by this Thesis.

On the other hand, multiview systems have multiple sources of video information; this means they code video data coming from multiple cameras. In this context, multiple bitstreams are created, typically one per camera; these bitstreams converge to the decoder which processes all the data, exploiting the correlation between the various cameras, so-called inter-view correlation, as expected by the DVC paradigm. Although the DVC multiview systems exist in smaller numbers in the literature, they effectively bring a significant architectural advantage over predictive multiview codecs as there is no need for the video cameras to communicate among them to exploit the inter-view correlation at the encoder side. For example, a

cluster of cameras used for security protection of some location, using a predictive coding design, may be rather expensive and complex, notably in terms of equipment and processing power, as the cameras have to communicate with each other and converge at the encoder side for posterior delivery to the decoder. The same problem would disappear in a DVC multiview set up, as the communication between cameras would not be needed, since the bitstreams coming from each camera would only converge to the decoder, where all the inter-view correlation should be exploited. In some systems, power consumption problems and bandwidth limitations are a serious problem that can be overcome by using a DVC multiview approach. One of the barriers preventing a more common use of these DVC multiview systems is the gap in RD performance regarding predictive multiview codecs, notably regarding the recently finalized Multiview Video Coding (MVC) standard [9]

2.1.2. Second Dimension: Low Delay versus High Delay

Considering the importance of delay issues for most video applications, it is rather natural that one of the classification dimensions relates to delay, notably distinguishing between low delay and non-low delay systems. A low delay system is required by applications, which cannot afford a delay higher than a specified value, notably interpersonal communications or surveillance applications; an extreme case of non-low delay is off-line coding where all the content is available before starting coding. The low delay requirement has a strong impact on the range of coding tools that may be used since some of them 'buy' RD performance with coding delay, e.g. like B frames in predictive video codecs. There are trade-offs pertaining the use of a specific set of tools in terms of RD performance and delay requirements. It is important to note that this delay does not refer to the computational delay, which may be reduced by using a more powerful platform, but rather to algorithmic delay, e.g. using a frame in the future, which does not depend on the available computational power. Since the 'high delay' systems do not have strong restrictions in terms of delay requirements, there is more freedom on the tools that may be used; this also typically implies more complexity. It is important to note that complexity is always a moving target, meaning that solutions that are today very complex will always be (relatively) less complex in the future due to Moore's law.¹

In terms of DVC systems, the delay is deeply related to the type of frame estimation solution used at the decoder when creating the so-called side information; while low delay usually asks for extrapolation solutions where future frames are not used, higher delays typically allow interpolation solutions using past and future decoded frames.

2.1.3. Third Dimension: Frame Based versus Block Based

There are many DVC solutions that use either a frame based or a block based approach as the main spatial support for the coding process and thus the creation of the coded bitstream. Although even frame based solutions may use block level processing at same stage, the final bitstream regards the frame since the bits cannot be individually and easily associated to a smaller spatial support like the block. While block level coding has the intrinsic advantage of exploiting the local features of the video, and thus being more adaptive and eventually efficient, frame based coding has the advantage to deal with larger portions of data which brings efficiency advantages for some channel codecs which play a paramount role in DVC. The most

¹ Moore's law describes a long-term trend in the history of computing hardware. Since the invention of the integrated circuit, the number of transistors that can be placed inexpensively on an integrated circuit has increased exponentially, doubling approximately every two years. Almost every measure of the capabilities of digital electronic devices is linked to Moore's law: processing speed, memory capacity, even the number and size of pixels in digital cameras.

important example of a frame based DVC solution is the Stanford DVC codec proposed in [1]. This architecture has been largely adopted by the DVC research community and, thus, there are many DVC solutions available which derive from this initial DVC approach; see Section 2.2.1 for details on a state-of-the-art DVC solution derived from the initial Stanford architecture. On the other hand, the most important example of a block based DVC solution is the Berkeley DVC codec, also known as PRISM from ‘Power-efficient, Robust, High-compression, Syndrome-based, Multimedia coding’ [7] and [8]. In this solution, the whole process begins with block classification to exploit the video local characteristics and the final bitstream represents each block, thus allowing the application of different coding approaches at the block level depending on the different amount of correlation present in the video data.

It is important to note that even if other major differences between these two initial DVC approaches exist (including the exploitation or not of a feedback channel), they have been smoothed over the years, with the development of DVC solutions combining elements from both initial DVC solutions.

2.1.4. Forth Dimension: Feedback Channel versus No Feedback Channel

The presence of a feedback channel in the DVC architecture is a very important feature since it has a strong impact on the codec’s behavior, notably in terms of rate control. The availability of a feedback channel allows a ‘tighter’ control of the bits sent from the encoder to the decoder, depending on the needs detected by the decoder since in DVC systems this is the ‘side’ performing the intensive processing. For example, if the encoder sends X parity bits through the communication channel but the decoder needs more bits to achieve a certain error probability, and thus the target video quality, it can use the feedback channel to request more bits from the encoder instead of providing decoded video with strong coding artifacts. This capability has certainly advantages but has also disadvantages. For example, this type of architectural approach can only be used in real-time applications since the amount of bits needed can only be known when the decoding is being performed (and it depends on the decoder) and, thus, the bits cannot be stored in advance as in video storage applications. Moreover, providing low delay in the context of real-time applications may be a problem, as the intensive use of the feedback channel and the associated delay may render this type of codec useless in this scenario. In this situation, it is necessary to create very good side information (the estimations of the original frame created by the decoder) to limit the usage of the feedback channel, thus reducing the amount of decoder requests for more bits to correct side information errors. Naturally, there are also other approaches to address this problem, notably eliminating the use of the feedback channel and adopting an encoder rate control approach or even a hybrid rate control approach where the encoder has the task to estimate with reasonable accuracy the amount of bits needed to achieve a certain error probability, and thus video quality, at the decoder [10], [11]. The best examples of these two types of DVC architectures, one with feedback channel and another without, are again the Stanford [1] and PRISM DVC solutions [7], respectively.

2.2. Most Relevant Distributed Video Coding Solutions

In this section, four state-of-the-art DVC solutions are reviewed. While some of them are directly related to the problem this Thesis is addressing, notably low delay DVC solutions, others are not but their results demonstrate the best advances made in DVC and their relevance justifies their presentation here.

2.2.1. The Transform Domain DISCOVER DVC Codec

2.2.1.1. Objectives

The main objective of the DISCOVER European project [12] was to provide a DVC codec – the DISCOVER DVC codec – adopting and improving the Stanford architecture and providing an increased RD performance, notably making further steps in closing the gap between the RD performance of DVC and predictive video coding systems. Although the DISCOVER DVC codec may be easily adapted for low delay performance, its current version does not address this functionality. For low delay performance, the DISCOVER DVC codec shall adopt a side information creation process based on extrapolation and not based on interpolation of past and future decoded frames, as it is currently the case. In the context of the classification tree proposed in Section 2.1, the main DISCOVER DVC codec is a monoview, non-low delay, frame based and feedback channel based DVC codec; the DISCOVER project has also a multiview DVC codec which will not be presented here [13]

2.2.1.2. Approach and Architecture

The DISCOVER DVC codec is based on the Stanford architecture as briefly presented in Chapter 1 [1]. However, new modules have been added to the architecture and some of the initial modules have also been improved, notably targeting to increase the RD performance. Some of the improvements are related to the replacement of the turbo codec by a Low-Density Parity-Check Accumulate (LDPCA) codec in the Slepian-Wolf part of the codec, the introduction of a Cyclic Redundancy Check (CRC) codec for error detection, the improvement of the Correlation Noise Modeling and Reconstruction modules, and the usage of the state-of-the-art H.264/AVC Intra codec for coding the key frames. The other modules remain basically the same. Figure 2.2 shows the entire DISCOVER DVC codec architecture with the exception of one of the modules (Frame Classification) described below.

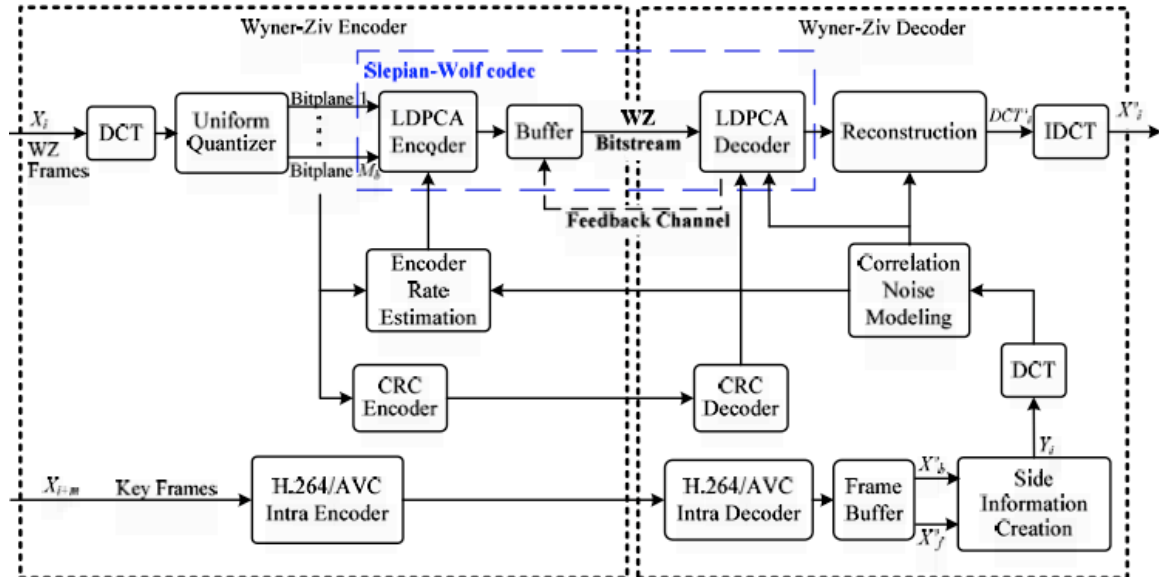


Figure 2.2 – DISCOVER DVC video codec architecture [20].

In the following, a walkthrough of the encoding and decoding processes is presented.

Encoding Process:

- **Frame Classification** – The first step in the encoding process consists in a simple classification of the video frames as key frames (KF) or Wyner-Ziv (WZ) frames. This choice is made by a module that is not present in Figure 2.2 and dynamically determines the GOP size, i.e. the number of WZ frames between two key frames depending on the amount of temporal correlation in the video sequence [30]; most DVC results in the literature use a fixed GOP size of 2 frames.
- **H.264/AVC Intra Encoding** – This module is responsible for encoding each key frame using an Intra codec to limit the encoding complexity. Depending of the GOP size used, and thus on the number of key frames, more or less bits are sent to the H.264/AVC Intra decoder.
- **Integer Discrete Cosine Transform** – To each WZ frame, an integer 4×4 block-based integer DCT is applied. Afterwards, the DCT coefficients of the entire WZ frame are grouped together, forming the DCT coefficient bands. This grouping is performed based on the position occupied by each DCT coefficient within the 4×4 blocks.
- **Quantization** – After the integer DCT is applied, each coefficient band b_k , resulting from the previous module, is uniformly quantized using 2^{M_k} levels. These 2^{M_k} levels depend on the visual sensitivity to each DCT coefficient band b_k . The result is a symbol stream where bitplane extraction is performed; the quantized symbols of the same significance are grouped together forming the corresponding bitplane array for posterior LDPCA encoding.
- **LDPCA Encoding** – The bitplane arrays resulting from the quantization process arrive at the LDPCA encoder where all the parity information is generated for each bitplane, stored in a buffer and sent to the decoder when requested through the feedback channel.
- **Encoder Rate Estimation** – The objective of this module is to adopt a hybrid rate control approach to limit the number of data requests made by decoder, decreasing at the same time the transmission delay and the decoding complexity. In this context, this module performs an initial estimation of the number of bits to be sent to the decoder for each bitplane, targeting a certain decoded quality. If there is underestimation, the decoder simply asks for more bits by making one or more requests through the feedback channel.
- **CRC Encoding** – A CRC-8 (extra 8 bits) checksum is transmitted to the decoder to help achieving a vanishing error probability for each decoded bitplane.

Decoding Process:

- **Side Information Creation** – This module is one of the most important in this entire DVC architecture since its performance strongly determines the usage of the feedback channel and the final RD performance. By adopting a motion compensated frame interpolation framework based on past and future reference decoded frames, the side information creation process can provide an estimation of the original WZ frame to be encoded. The better the estimation, the smaller is the number of requests made by the decoder to correct the estimation errors in the side information; as a consequence, it lowers the necessary bitrate for successful decoding.
- **DCT Estimation** – By performing a 4×4 block-based DCT over the created side information, an estimation of the WZ frame DCT coefficients is obtained.

- **Correlation Noise Modeling** – A Laplacian distribution is used to model the residual statistics between the original WZ frame DCT coefficients and the side information DCT coefficients. Note that the Laplacian parameter is estimated online, at different granularity levels, notably at band and coefficient levels [14], [15].
- **LDPCA Decoding** – As soon as the LDPCA decoder receives information for the residual statistics for a given DCT coefficients band b_k and the DCT transformed side information, the decoded quantized symbol stream associated to the DCT band b_k can be obtained through the LDPCA decoding procedure applied to successive chunks of parity bits received from the encoder, following each decoder request through the feedback channel. The LDPCA starts by decoding the most significant bitplane array of band b_k , proceeding in an analogous way to the next bitplanes M_{k+1} in each band. Once all the bitplanes have been successfully decoded for a band, the LDPCA decoder proceeds to the next band b_{k+1} . This process stops when all the DCT coefficients/bands have been decoded.
- **Request Stopping Criterion** – The decoder checks if all the LDPCA code parity-check equations are fulfilled for the decoded codeword after decoding each ‘chunk’ of parity bits received from the encoder. If that is the case, it means that no more bits are needed to decode that bitplane, hence the decoding process for the next bitplane can start; otherwise, the decoder requests for more parity bits through the feedback channel.
- **CRC Checking** – As mentioned before, a CRC checksum is transmitted to help the decoder detecting and correcting the remaining errors in each bitplane. As the CRC combines efforts with the request stop criterion, it does not have to be very complex/robust to achieve a vanishing error probability for each decoded bitplane.
- **Symbol Assembling** – The bitplanes resulting from LDPCA decoding the M_k bitplanes associated to the DCT band b_k are grouped together to form the quantized symbol stream. The DCT coefficient bands, also called *unquantized* bands, for which no WZ bits were transmitted, are replaced by the DCT coefficient bands directly resulting from the side information process.
- **Reconstruction** – As soon as all the quantized symbols are obtained, the matrix with the decoded DCT coefficients for each block is reconstructed.
- **IDCT** – Next, it is necessary to perform an inverse 4×4 block-based transform to obtain the WZ frame in the pixel domain.
- **Frame Remixing** – To get the final video sequence, both the key frames and WZ frames have to be properly (temporally) mixed.

2.2.1.3. Main Tools

There are several modules in the DISCOVER DVC codec that can be considered main tools such as the Transform and Quantization, Slepian-Wolf Coding, Side Information Creation, Correlation Noise Modeling and, finally, the Reconstruction.

A) Transform and Quantization

The **Transform and Quantization** is the first coding stage of WZ frames as the process begins by transforming a WZ frame using a 4×4 block based integer DCT transform, from left to right, and top to bottom, forming the coefficients bands b_k . The resulting DC and AC coefficients have different treatment in terms of quantization. The DC coefficients are characterized by high amplitudes as they express the average energy of the sampled 4×4 block and, thus, are quantized using

a uniform scalar quantizer without a symmetric quantization interval around the zero amplitude. The AC coefficients are also quantized using a uniform scalar quantizer with even quantized bins, evenly distributed around zero to prevent the block artifacts effect [17]. Note that the bin around zero is twice the size of the other bins; as a consequence, the matching probability between the quantized bins of the WZ and SI frames increases, lowering the bitrate. In compensation, some distortion loss is expected since the quantization bin is bigger and the decoded frame quality is lower; however, overall, the RD performance improves. Each AC coefficient band has a varying quantization step size, allowing an adjustment depending on its dynamic range. The step size, W , for the AC band is obtained from $W = 2|V_k|_{\max} / (2^{M_k} - 1)$, where $|V_k|_{\max}$ corresponds to the highest absolute value in the band k .

B) Slepian-Wolf (SW) Coding

A class of LDPC codes, more specifically a LDPC syndrome code, concatenated with an accumulator, performs the **Slepian-Wolf (SW) Coding** stage [18]. The result is an adaptive and incremental code that allows the use of the request-decode strategy, as the produced accumulated syndromes are stored in the encoder buffer and transmitted to the decoder upon its requests. For each bitplane, a syndrome s is calculated based on a parity check matrix H . This matrix represents the connections of a bipartite graph with two node types, the variable nodes and the check nodes. Due to its low density of 1's, the parity check matrix guarantees a low encoding complexity. The LDPC syndrome decoder attempts the reconstruction of the original data with the help of the available side information and the correlation noise model, using an adapted Sum Product Algorithm (SPA). If the number of syndrome bits is not enough, SW decoding fails and the encoder is notified through the feedback channel that more accumulated syndromes are needed. The criterion enabling this type of decision is based on a convergence test involving the computation of the syndrome check error. Basically, there are a number of parity-check equations that need to be satisfied in order to proceed to the next step. Keeping in mind that the syndrome check error is computed for up to 100 iterations when LDPC decoding, and the objective is to have this error approaching zero within that number of iterations, two outcomes are possible. If after 100 iterations the syndrome check error still remains different from zero, the solution lies in requesting more accumulated syndromes from the encoder. If within that number of iterations the error turns out to be zero, then the next step is triggered, where the received CRC-8 checksum is compared to the corresponding bitplane decoded CRC checksum. In case the checksums are the same, the decoding is successful and the decoding of another band/bitplane can start. When all the syndromes bits are received, then the source is recovered by using an inverse H matrix. This recover can be lossless as this matrix is full rank and the LDPC syndrome code is linear.

C) Side Information Creation

The side information creation is the most important stage in the DVC codec, as its quality limits the number of errors which need to be corrected through LDPCA syndrome bits. Intuitively, the better the quality of the side information, the lesser the need for error correction through the LDPCA decoder, increasing at the same time the codec's RD performance. The side information creation process can be performed in many different ways. The DISCOVER DVC codec uses a motion compensation interpolation framework based on both the local motion associated to the moving objects and also the general motion of the scene. Interpolation suggests the use of a past reference frame and a future reference frame. Depending on the GOP size, the frames used vary; for example, if the GOP size is 2, the past and future frames will be both key frames. For

other GOP sizes, other estimation structures are adopted, including the use of decoded WZ frames. In the DISCOVER DVC codec, the side information creation process includes several steps, notably:

- **Low Pass Filtering** - The past and future decoded reference frames are first low passed filtered to improve the accuracy of the motion vectors. Then, to estimate the motion between the two reference frames, a block-matching algorithm is used. This algorithm, proposed in [30], involves full motion estimation with a modified matching criterion.
- **Motion Vectors Refinement** - To refine the motion vectors obtained in the previous step, a bidirectional motion estimation process is applied, taking into account an additional constraint: the motion vector selected for each block must have a linear trajectory between the two reference frames, crossing the interpolated frame at the center of the blocks. At the same time, the hierarchical block size motion estimation technique, proposed in [16], is performed, based on a coarse-to-fine approach, tracking fast motion and handling larger blocks (16×16) in the first iteration, and then progressing to smaller block sizes (8×8), if needed.
- **Motion Field Smoothing** – Next, an algorithm using spatial motion smoothing [30], based on weighted vector median filters, is used; the purpose is to create a smoother final motion vector field, with the exception of the object boundaries and uncovered regions.
- **Motion Compensation** - As soon as the final motion vector field is available, the bidirectional motion compensation is applied, resulting in an interpolated frame, ready to be DCT transformed.

Figure 2.3 shows the Frame Interpolation Framework architecture, which provides the side information frames in the DISCOVER DVC codec.

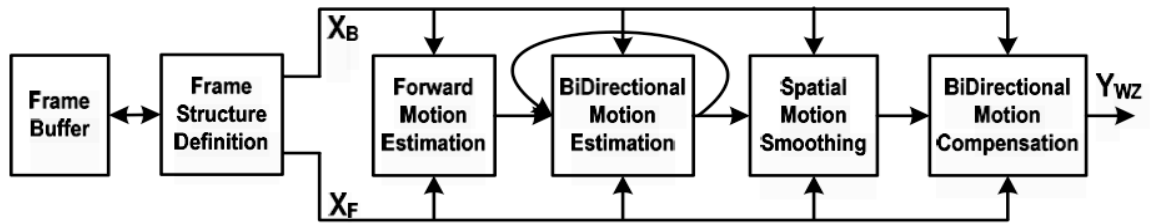


Figure 2.3 – Frame Interpolation Framework architecture [20].

D) Correlation Noise Model

The correlation noise model is responsible for using the side information to provide the SW decoder information on the probabilities of the various bits for each bitplane. For that to happen, a Laplacian distribution is adopted to model the residual statistics between the DCT coefficients of the quantized original WZ frame and the DCT coefficients corresponding to the side information. It is important to note that the Laplacian distribution α parameter is estimated online, providing a more realistic solution in comparison with many codecs based in the Stanford architecture that use offline estimation [1].

E) Reconstruction

The reconstruction process is the final decoding stage, gathering the LDPC decoded bitplanes, the side information and the residual statistics for each DCT coefficient band to obtain the decoded DCT coefficients matrix [19]. To obtain each DCT coefficient, the bitplanes are grouped to form a band of DCT coefficients. Then, for each band, a decoded quantization symbol q' is obtained, each corresponding to a DCT coefficient, indicating the quantization interval where the original DCT lies. Considering x' to be the reconstructed DCT coefficient, it is necessary to define the reconstruction function responsible for obtaining that value:

$$x' = E[x | q', y] = \frac{\int_l^u x \cdot f_{x|y}(x | y) dx}{\int_l^u f_{x|y}(x | y) dx} \quad (2.1)$$

After analytical manipulation of (2.1) the reconstructed value x' is obtained with the following equation:

$$x' = \begin{cases} l + b & , y < l \\ y + \frac{\left(\gamma + \frac{1}{\alpha}\right)e^{-\alpha\gamma} - \left(\delta + \frac{1}{\alpha}\right)e^{-\alpha\delta}}{2 - e^{-\alpha\gamma} - e^{-\alpha\delta}} & , y \in [l, u[\\ u - b & , y \geq u \end{cases} \quad (2.2)$$

where:

$$\begin{aligned} b &= \frac{1}{\alpha} + \frac{\Delta}{1 - e^{-\alpha\Delta}} \\ \gamma &= y - l \\ \delta &= u - y \end{aligned}$$

where Δ represents the quantization bin size and y represents the DCT coefficient of the side information. As defined in equation (2.2), only 3 situations may occur. When the DCT coefficient generated by the side information is below the lower bound l defined by the reconstruction function or above the upper bound u , x' is computed using only 2 more values: the quantization bin size and the α Laplacian parameter. In case y is within the boundaries, then y itself helps to calculate the reconstructed DCT coefficient. In either case, there is a trend to shift the reconstructed DCT coefficient to the center of the decoded quantization bin. Every DCT coefficient with no WZ bits associated is directly made the same as the side information DCT coefficient.

2.2.1.4. Performance Evaluation

The DISCOVER DVC codec performance evaluation has been done in several dimensions, namely RD performance, feedback channel rate performance, encoding complexity and decoding complexity [20]. Due to space constraints, only the RD performance results will be reported here; for more detailed information, please consult [20]. The test conditions used to evaluate the codec's RD performance are defined in Table 2.1 and in Figure 2.4.

Table 2.1 – DISCOVER test conditions.

Sequences	Foreman (with the Siemens logo), Hall Monitor, Coast Guard, Soccer
Frames	All frames for each sequence
Spatial Resolution	QCIF (176×144), only luminance
Temporal Resolution	15 Hz
GOP Length	2 (standard if no more information is provided), 4 or 8



Figure 2.4 – Sample frames for test sequences: a) Foreman (Frame 80); b) Hall Monitor (Frame 75); c) Coast Guard (Frame 60); d) Soccer (Frame 8).

In terms of quantization, eight matrices are defined including the number of quantization levels associated to the various DCT coefficient bands. In the following, each specific quantization matrix i , where $i=1,\dots,8$, will be referenced as Q_i .

The RD performance evaluation is done for the luminance information including both the WZ frames and the key frames. The DISCOVER DVC codec RD performance is compared with relevant benchmarks, notably several low encoding complexity predictive codecs (without encoder motion estimation), such as H.264/AVC Intra, H.264/AVC Inter No Motion and H.263+ Intra.

In general terms, the DISCOVER DVC codec has a promising RD performance behavior, being able to compete and typically perform better than the H.264/AVC Intra and H.263+ Intra codecs. For the sequence Soccer, where there is a lot of motion, the DISCOVER DVC codec performs below all the others codecs since the quality of the side information is rather poor. However, for all the other sequences, the DISCOVER DVC codec shows very similar RD performance to the recent H.264/AVC Intra codec, and even surpasses the H.264/AVC No Motion codec for the Coast Guard sequence.

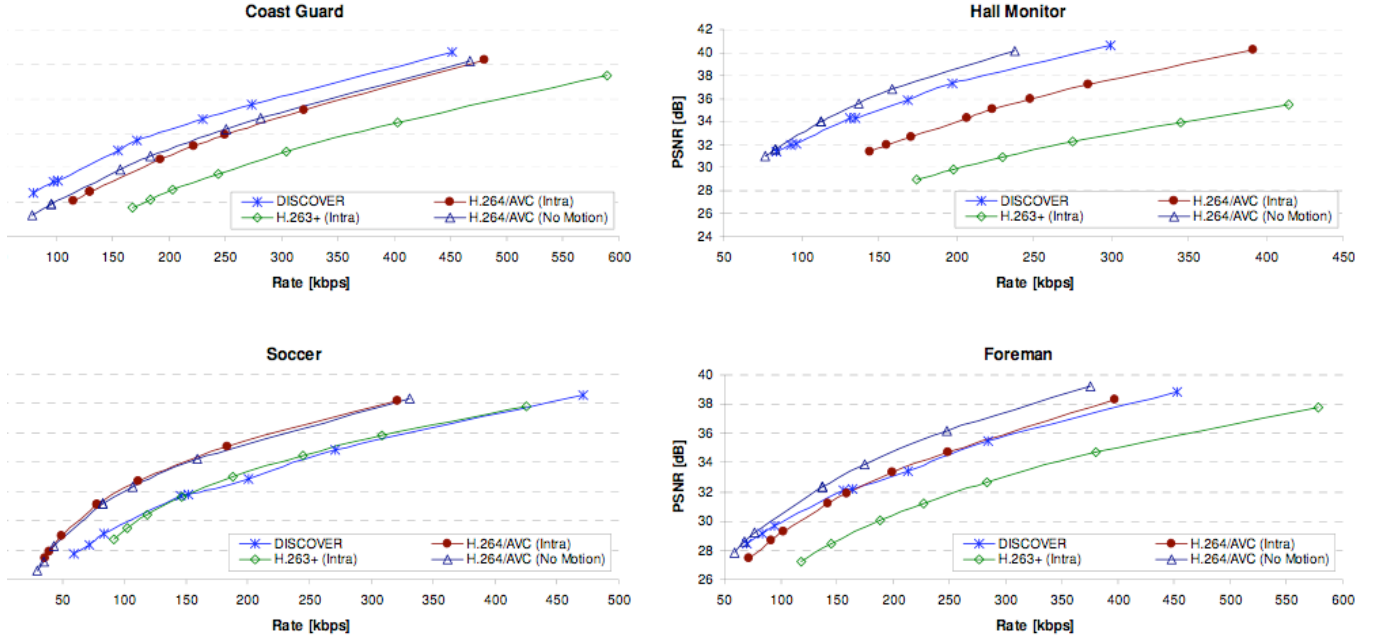


Figure 2.5 – RD performance comparison for various test sequences (QCIF, 15 Hz, GOP=2) [20].

2.2.1.5. Summary

The DISCOVER DVC codec presents promising RD performance results regarding the relevant standard codec alternatives, notably for short GOP sizes and low and regular motion video. However, it is not suitable for low delay applications as the side information creation process uses an interpolation framework adding undesired delay. It is important to note that many of the modules still have room for improvements which means that further RD performance gains may be expected in the future, further closing the gap between predictive and distributed video coding systems.

2.2.2. Low Delay Pixel Domain IST DVC Codec

2.2.2.1. Objectives

The objective of this codec, proposed by Natário et. al. from Instituto Superior Técnico (IST) in [24], is to provide a RD efficient low delay solution based on the Stanford DVC architecture [1]. With this target, the typical side information interpolation method is not an acceptable solution, as it requires the use of future decoded frames, thus increasing the algorithm delay and rendering the DVC solution useless for low delay purposes. Keeping this in mind, a new DVC solution based on side information derived through extrapolation methods is described below. Besides low delay, this solution also strives for low encoding complexity and the highest RD performance. Contrary to the previously described DISCOVER DVC codec, the Low Delay IST DVC codec is a pixel domain DVC solution meaning that no transform is applied to the WZ frames. According to the classification tree proposed in Section 2.1, this codec is a monoview, low delay, frame based with feedback channel DVC solution.

2.2.2.2. Approach and Architecture

The architecture presented in Figure 2.6 shows the same main modules as the architecture described in the previous section since they are both based on the initial Stanford DVC solution; as a consequence, the common modules will not be described in detail here. As a difference to the DISCOVER DVC Codec, the conventional Intra frame codec for the key frames uses the H.263+ standard instead of the more recent H.264/AVC standard. Moreover, the Slepian-Wolf codec uses turbo codes instead of LDPC codes. The side information creation process is based on a novel frame extrapolation scheme which constitutes the main difference between this codec and the one presented in the previous section.

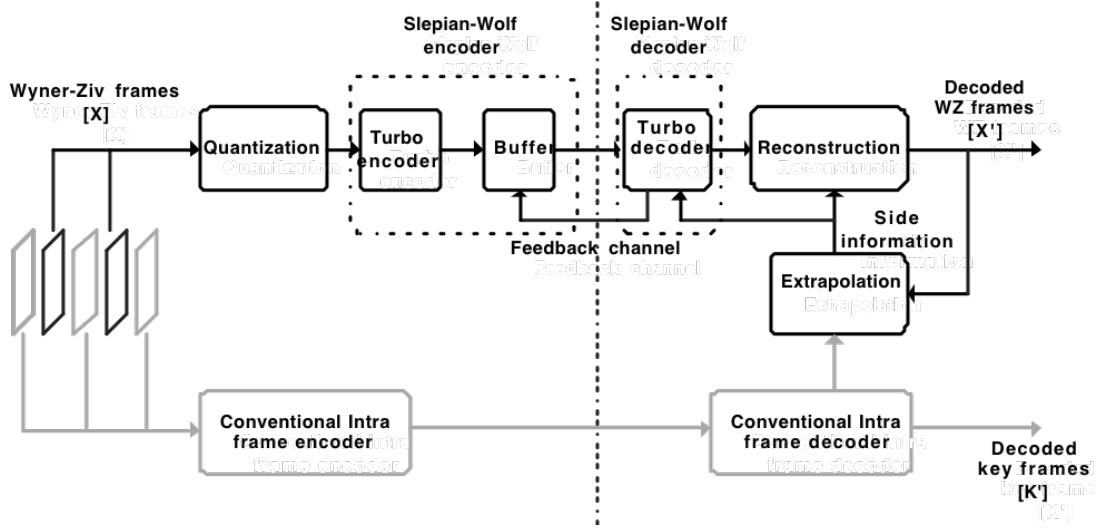


Figure 2.6 – Low Delay IST DVC codec architecture with side information extrapolation [24].

At the decoder, the most important process pertains to the frame extrapolation method for creating the side information as it only uses past decoded frames, both key frames and WZ frames depending on the GOP size. The side information generated serves not only for the turbo decoder to achieve the required error probability error for the various sample bitplanes but also for the reconstruction process. This extrapolation technique includes four steps as described in detail in the next section. Depending on the quality of the extrapolated side information, more or less bits have to be requested by the turbo decoder through the feedback channel. The final result is a sequence of WZ frames and key frames that represent the original video sequence with a certain target quality.

2.2.2.3. Main Tools

The main novel tool used in the Low Delay IST DVC codec, and also the most important one in the context of this Thesis, is the frame extrapolation method for side information generation. As already mentioned, the quality of the side information strongly determines the codec's RD performance. The better the extrapolation method, the better the side information, and thus the lesser the errors needing correction in the turbo decoder and, consequently, the lesser the bits requested over the feedback channel.

There are several ways to generate side information based on extrapolation schemes, but the one discussed in this section is based on motion projection. The stages that define the proposed extrapolation process are shown in Figure 2.7; their objective is to create a ‘high quality’ extrapolated WZ frame.

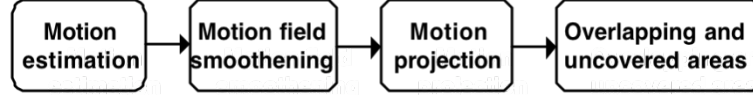


Figure 2.7 – Side information creation by extrapolation architecture [24].

- **Motion Estimation** - The purpose of the motion estimation step is to generate motion vectors based on 8×8 overlapped blocks, using the two previously decoded frames (both in the past). The overlapped blocks are used to reduce the block artifacts in the side information frame caused by block motion compensation.
- **Motion Field Smoothing** – As the name itself implies, this stage performs a smoothing of the obtained vector field by calculating a refined motion vector for each block, resulting from averaging all neighboring vectors.
- **Motion Projection** - By applying the motion field to the previous frame, an extrapolated frame is obtained, assuming the motion is linear, as shown in Figure 2.8. This is not the final frame yet, mainly because the treatment of the overlapped and uncovered areas is still undone.

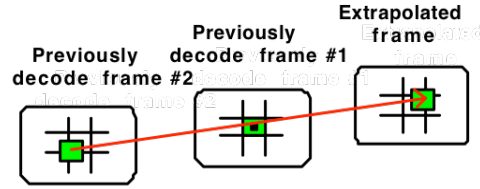


Figure 2.8 – Motion projection assuming linear motion [24].

- **Overlapping and Uncovered Areas** – Two situations may occur in the extrapolated frame resulting from the motion projection. The first is an overlapped area resulting from a pixel being estimated by more than one pixel in the previous frame; when that happens, the solution proposed is to use the average between the values of the various pixels overlapping. The second situation regards the areas for which no information is assigned and, thus, needs to be filled. In this case, a local spatial interpolation is performed, based on three neighbors (up, left and up-left), scanning the extrapolated frame from top to bottom and left to right.

2.2.2.4. Performance Evaluation

The evaluation of this codec is based on the RD performance. A first test compares several DVC architectures, namely the codec developed at IST with motion interpolation [16] (very similar to the DISCOVER DVC codec), the Low Delay IST DVC codec with motion extrapolation (described in this section), the transform domain Stanford DVC codec proposed in [23] and, finally, the standard H.263+ Intra mode codec. A second test evaluates the RD performance of the Low Delay IST DVC codec for various GOP sizes in comparison with the performance of the H.263+ Intra mode codec.

While the first test uses as test material the first 100 frames of the Foreman sequence, QCIF spatial, at 30 fps, GOP size 2, counting only the WZ frames bitrate, the second test uses QCIF spatial resolution, at 30 fps and the Galleon sequence from the Video Quality Experts Group (VQEG) test set.

The results associated with the first test are showed in Figure 2.9 a) and allow concluding that the IST DVC codec with motion interpolation [16] presents better RD performance results than the motion extrapolation based Low Delay IST DVC codec; however, the Low Delay IST DVC codec performs better than the solution in [23], which uses interpolation to generate the side information.

Regarding the second test, the results in Figure 2.9 b) for the Low Delay IST DVC codec show that increasing the GOP size induces, as expected a RD performance drop. At the same time, the RD performance for all tested GOP sizes, up to 100, is able to overcome the RD performance for the H.263+ Intra codec by a large margin.

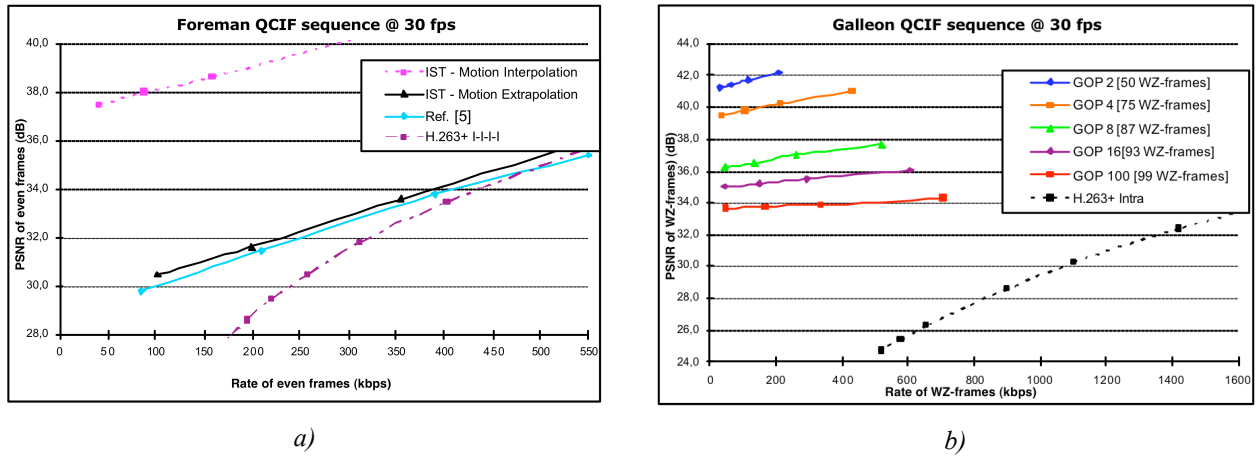


Figure 2.9 – RD Performance comparison between several GOP sizes for a) Foreman sequence, b) Galleon sequence [24].

2.2.2.5. Summary

The codec discussed in this section demonstrates the performance of a motion extrapolation scheme, for low delay applications, while maintaining low complexity at the encoder. Since only the bitrate of the WZ frames is considered for the performance evaluation, ignoring entirely the key frames bitrate, there is the impression that original key frames have been used. It is important to keep in mind that far more recent tools are available that can be used to improve the codec above, potentially boosting its RD performance; for example, the use of H.264/AVC Intra instead of H.263+ Intra and the use of LDPC codes instead of turbo codes could improve the RD performance. For more information about this codec, please read [24].

2.2.3. Pixel Domain Low Delay DVC using Iterative Refinement Side Information Generation.

2.2.3.1. Objectives

The objective of this codec developed by Weerakkody et. al. from University of Surrey [25] is to create an improved version of the Stanford DVC codec, with higher RD performance, using novel techniques regarding those described in previous sections. In order to achieve this objective, the authors use an iterative refinement technique to improve the initially

created side information, exploiting both the spatial and temporal correlations in video frames while simultaneously using the ideas presented in Section 2.2.2 for the Low Delay IST DVC codec [24]. This codec is monoview, low delay, frame based and uses a feedback channel. Note also that this is a pixel domain codec; for more information, please read [25].

2.2.3.2. Approach and Architecture

The codec architecture shown in Figure 2.10 is very similar to the codec architecture in the previous section with the exception of a new decoder module called Refinement, which corresponds to the major novelty of the codec presented in this section; thus, the parts of the codec which are similar in previously described codecs will not be described here to avoid repeating what has already been said before. The key frame's codec is not explicitly mentioned in [25] but any of the Intra codecs available and used in the previously presented solutions could be used here. The Slepian-Wolf coding is done with turbo codes as in the original Stanford DVC solution.

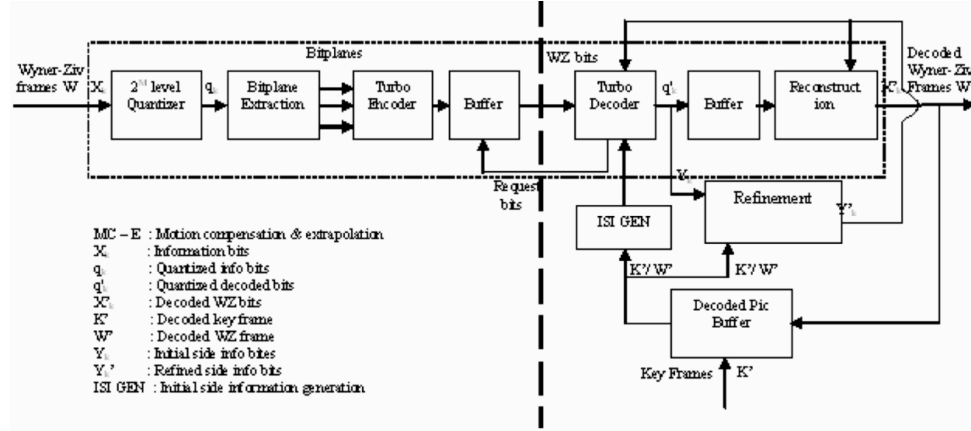


Figure 2.10 – DVC architecture for the iterative refinement technique for side information generation [25].

Since the encoding process is similar to the encoding processes described in previous sections, only the main decoding modules are presented here.

- **Initial Side Information Generation (ISI GEN)** – This module is responsible for the creation of the initial side information; it uses a motion extrapolation method based on the same SI generation process presented in Section 2.2.2 [24].
- **Refinement** – The refinement of the side information is based on an iterative process including several steps to achieve at the end a better and refined side information frame. The basic intuition for this refinement process is to detect and ‘improve’ parts of the initial side information frame which do not seem ‘consistent’ and thus behave as ‘outlier’ areas, thus reducing the RD performance. Further details are presented in the section below.

2.2.3.3. Main Tools

As expected, the main tool used in this codec is related with the side information generation process, using as starting point the side information extrapolation technique proposed by Natário et al. in [24] and presented in the previous section. The proposed process increases the decoder complexity, adding loops and new steps for processing the initial side

information, as shown in Figure 2.11. The various modules in the proposed iterative refinement side information generation process are described below.

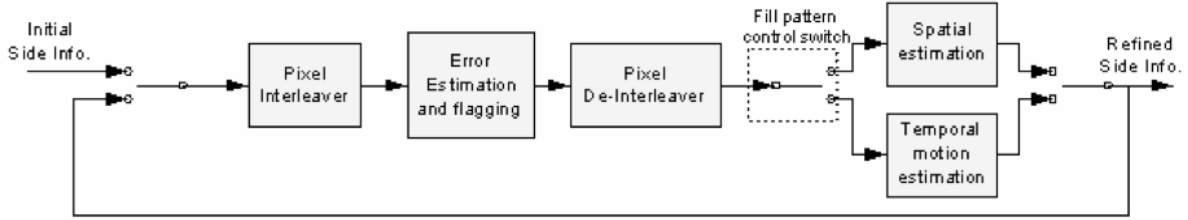


Figure 2.11 – Proposed iterative refinement side information generation architecture [25].

A) Side Information Iterative Refinement

- **Pixel Interleaver** – The initial side information frame is interleaved for better error protection, namely to prevent the typical burst errors that often happen. This interleaving process is no more than a simple scattering of the pixels belonging to the side information frame which helps to suppress the adverse effects of burst errors typically observed in the side information stream by scattering the bit sequence.
- **Error Estimation and Flagging** – The next step begins with the processing of chunks of bits where each chunk has length of 1×10 bits (experimental value). Then, the errors are estimated and compared with pre-determined error threshold used by other DVC codecs, presented in [25]. If the error rate calculated for that chunk is above the threshold defined, then all the bits belonging to that particular chunk are flagged as good.
- **Pixel De-Interleaver** – After performing the previous step, the side information frame is de-interleaved causing the flagged bits to scatter.
- **Filling Bits in Frame** – Using the fill pattern control switch to choose between spatial estimation and temporal motion estimation, an algorithm is performed to fill a frame with the flagged bits. Unfortunately, no more information is given by the authors about the spatial estimation and temporal motion estimation blocks and their corresponding algorithms.

The whole refinement process is repeated several times; through experimentation, it has been concluded that the optimum result is achieved by initiating the process with spatial estimation and performing, at least, 4 iterations. As soon as the refinement is completed, the resulting side information is sent to the turbo decoder to help decoding the parity bits received. After bit error rate (BER) estimation in the turbo decoder, two situations may occur: i) the resulting bit error rate is above the defined error rate threshold and then more parity bits are requested through the feedback channel, as many times as needed until the BER is achieved; ii) otherwise, the decoding is stopped and the decoder output bits are sent to the reconstruction module.

It is important to note that this codec uses at the same time two more tools to help generate better side information. The first tool, proposed by same authors in [27], allows the creation of two side information frames using either two key frames or a key frame and a Wyner-Ziv frame. Depending on the type of sequence, notably high or low motion content, one frame or the other is chosen and sent to the turbo decoder to help the decoding process. Regarding the side information sent to the reconstruction block, it is always the same to avoid discontinuity problems, notably the frame using two key frames to

extrapolate the current frame. The second technique is proposed in [28] and corresponds to a sequential motion estimation process based on the luminance and chrominance components.

2.2.3.4. Performance

The performance of this codec is evaluated in similar test conditions as the Low Delay IST DVC codec, notably using the first 100 frames of the Foreman sequence. The granularity of the quantizer can be chosen between 4 values, by manipulating M , where $M=1,2,3,4$. The WZ frame rate is 15 fps, and the results correspond only to the luminance component. The number of refinement iterations performed to achieve an optimum result was 4.

Keeping in mind that the PSNR results shown in Figure 2.12 represent an average of the tests for the different granularities of the quantizer, the proposed algorithm performs better in terms of RD performance than the alternative DVC solutions proposed in [24], [26], [27] using extrapolation methods and the solution in [28] that does not use extrapolation. These results come at the cost of some additional decoding complexity corresponding to the novel refinement module as well as the use of multiple side information streams.

2.2.3.5. Summary

Even though the proposed low delay DVC codec increases the RD performance at the cost of some additional decoding complexity associated to the refinement process, the encoding complexity is not changed. Considering the results presented in Figure 2.12, the proposed extrapolation technique combines multiple tools, which seem to show promising performance.

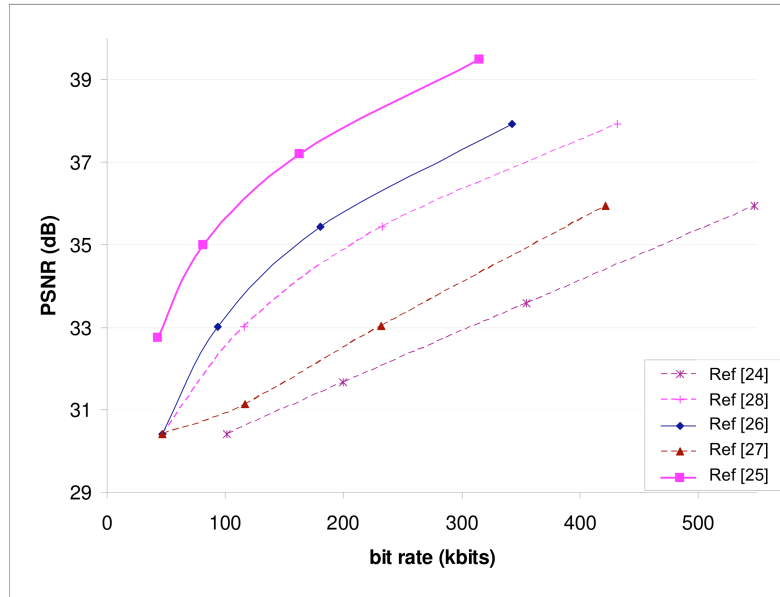


Figure 2.12 – PSNR comparison between several codecs [25].

2.2.4. Motion Compensated Extrapolation Techniques for Side Information Generation

2.2.4.1. Objectives

The main objective of the extrapolation approaches proposed by Borchert et. al. in [21] and [22], is to reach good quality side information using several extrapolation techniques for the side information generation process. Thus, this section relates only to a specific, although very important, part of a low delay DVC codec. The approaches presented target low delay codecs; regarding the other classification dimensions related to the classification tree in Figure 2.1, they can be used both in multiview as monoview systems, as well as with and without feedback channel. In terms of the block based versus frame based dimension, these approaches are more common in frame based coding systems.

2.2.4.2. Approach and Architecture

Even if the extrapolation approaches to be presented do not constitute an entire codec, they are an important part. Keeping that in mind, examples of possible architectures using these extrapolation techniques are the architectures presented in the previous sections. Of course, these solutions can be used for both pixel and transform domain DVC codecs.

2.2.4.3. Main Tools

This section will present several frame extrapolation solutions. These solutions use a couple of basic motion estimation tools, which will be presented first.

A) Basic Motion Estimation Tools

The 3-D Recursive Search (3DRS) algorithm proposed in [34] is used to build sets of motion vector candidates with the help of spatio-temporal predictions. Some considerations have to be taken in account, namely the fact that the objects in pictures are typically bigger than the blocks; this makes the vector estimated for the neighbor block a very good candidate for the current block. The block scanning occurs from left to right, top to bottom; the blocks already scanned at a certain point are called *spatial candidates*, while the others that have to be taken from previous scans are called *temporal candidates*; Figure 2.13 shows this situation.

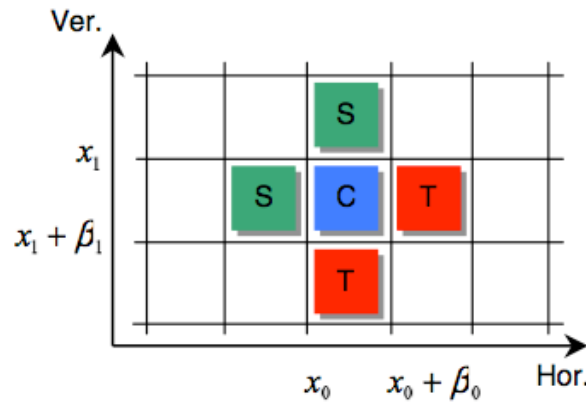


Figure 2.13 – Building the candidate set: *C* is the current block while *S* and *T* represent the spatial and temporal candidates [34].

Along with the candidates defined above, another one called *update candidate* is added to the set; this is a small random vector update that bases its computation on the following assumption: all objects have inertia. Consequently, this update vector is small, as the objects move continuously and not that fast from frame to frame. After computing all these candidates, a set is build containing two spatial, one temporal and one update candidates. Finally, to find the true motion vector, a Sum of Absolute Differences (SAD) is computed to find the vector that minimizes the matching error.

The Content Adaptive Recursive Search (CARS) algorithm presented in [34], is another algorithm used in the frame extrapolation approaches presented in the following. Motion estimation has three major problems, namely occlusion, aperture problem and sensitivity to noise. The block size is directly related with the latest two, as big blocks tend to reduce them. However, spatial accuracy is needed in several occasions, thus the need for smaller blocks. Consequently, different block resolutions such as 32×32 , 16×16 , 8×8 , 4×4 and 2×2 can be used in this algorithm allowing adjustment to different situations. This method performs 3DRS for each resolution creating different candidate sets. For example, 32×32 blocks are appropriate for large areas that have equal motion. Also candidates resulting from a global motion parametric model are added to this resolution. Update candidates can be used at the end to compensate small variations from the global motion model. In lower resolutions, such as 16×16 , instead of resorting immediately to the update candidates, spatial and temporal candidates in the vicinity are first tried out. The lower resolutions, such as 8×8 , are used to increase the spatial accuracy, usually associated with block positions that represent edges between objects. In most cases, one object has a spatial candidate and the other object has a temporal candidate. It is important to note that update candidates can always be used to increase the precision of the motion vectors, if needed.

B) Frame Extrapolation using a Forward Motion Estimation Approach

There are two variations pertaining forward motion estimation proposed by the same authors, which have similar processing steps.

The First Variation for Forward Motion Estimation (FVFME), is proposed in [22] and may be divided into five steps, using three previously decoded frames as input to the extrapolation algorithm:

- Generate a motion vector field for position $n-2$, i.e. estimate motion between frames $n-2$ and $n-3$, as well as between frames $n-2$ and $n-1$. As this process uses three frames, it is called a three frames motion vector field (3 frames ME).
- Generate a vector field for position $n-1$, i.e. estimate motion between frames $n-1$ and $n-2$. As this process uses two frames, it is called a two frames motion vector field (2 frames ME).
- Find areas addressed more than once, i.e. check the consistency of the vectors by shifting the field generated in step 1 to frame $n-1$. By comparing the shifted vectors applied in frame $n-1$ and the ones resulting from step 2, better results are achieved by choosing the vector with the lowest difference.
- Perform motion compensated extrapolation of the vector field, taking step 3 into account. Fill the unreferenced areas with temporally previous vectors (temporal hole filling), i.e. copy the vector at the coinciding position from vector field in step 2.
- Extrapolate current motion compensated frame.

The same authors propose in [21] a Second Variation for Forward Motion Estimation (SVFME). The main differences between these two forward motion estimation variations lie on the detection of uncovered areas, vector field analysis to

correct the vector pointing to unreferenced areas and the usage of spatial hole filling instead of temporal hole filling. The steps are almost the same in the beginning, namely the first two steps, but become different towards the end; a visual aid can be seen in Figure 2.14:

- Generate three frames motion vector field for position $n-2$ (estimate motion between $n-2$ and $n-3$, as well as between $n-2$ and $n-1$)
- Generate two frames motion vector field for position $n-1$.
- Use the 3DRS algorithm in step 2 to achieve better results for foreground vector in background region A, seen in Figure 2.14.
- Vector pointing from A to B will have a poor match and thus is invalid.
- Detect uncovering areas using the three frames motion estimation algorithm.
- Analyze vector field and correct vectors that point to unreferenced areas either with a neighbor vector or a vector from the three frames motion estimation.
- Find areas addressed more than once (C) and check consistency of the vectors.
- Extrapolate current motion compensated frame.

Apply spatial hole filling by iteratively computing the average value of the neighbor pixels.

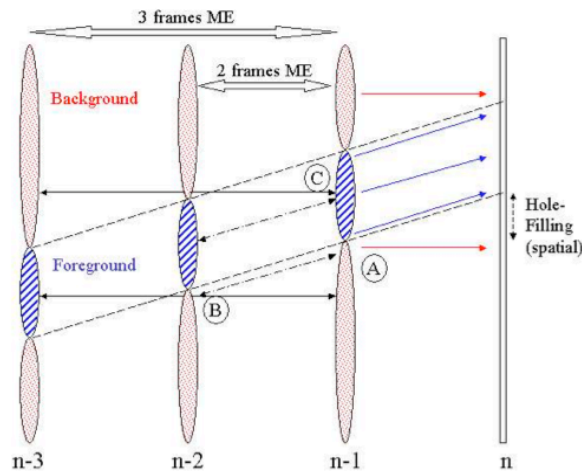


Figure 2.14 – Extrapolation scheme for forward motion estimation [21].

C) Frame Extrapolation using a Backward Motion Estimation Approach

There are other approaches that can be adopted to design a good frame extrapolation scheme [21], notably approaches like backward motion estimation, shown in Figure 2.15 presenting, like the forward motion estimation approach, different variations to improve the results. In this situation, the vectors point to the previous frame instead of to the next frame. Taking this into account, the current frame n is obtained by shifting the vectors from frame $n-1$ to frame n . This can generate a series of problems, being the most important one, how to choose the right vector. To overcome this problem, two solutions are proposed:

- **Simple backward motion estimation (SBME)** - Assumes the last vector found for a certain position is always the correct one, thus overwriting the one previously selected.

- **Improved backward motion estimation (IBME)** - Having a three frames vector field for frame $n-2$ and a two frames vector field for frame $n-1$, the solution passes by retiming the three frames vector field for frame $n-2$ to the frame $n-1$ itself and analyzing the collisions. The vector more similar to the two frames vector field at the position it points in frame $n-1$ is the best candidate.

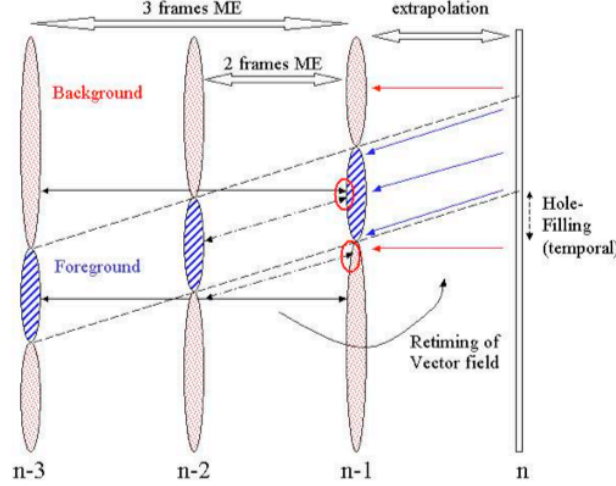


Figure 2.15 – Extrapolation scheme for backward motion estimation [21]

In terms of hole filling, this approach uses temporal correlation instead of spatial correlation, i.e. given a specific position of the current frame, if no vectors are appointed, the solution is to copy the corresponding vectors regarding those same positions but from the previous frame.

D) Frame Extrapolation using a Bilateral Filtering Approach

One last relevant frame extrapolation approach, still using backward motion estimation, adds one final step, so-called *bilateral filtering*. Bilateral Filtering (BA) [29] is a combination of domain and range filtering. Domain filtering is nothing more than traditional filtering as it weights the pixel values with coefficients that fall off with distance. Range filtering averages the pixel values with weights that decay with dissimilarity. It is important to note that bilateral filtering is applied not to the frame itself but to the final motion field that is created with the frame extrapolation approach. For further information, please check [21] and [22].

2.2.4.4. Performance Evaluation

As several frame extrapolation solutions were proposed and tested, it is necessary to define the set of conditions for which the tests were performed. As seen above, the first (forward) approach with two variations and second (backward) approach with three variations are properly identified.

The FVFME test uses QCIF (176×144) spatial resolution and several test sequences as shown in Table 2.2. The PSNR, corresponding only to the side information generated using the original key frames, is averaged, i.e. by modifying the step size of the quantization. The SVFME test uses a different spatial resolution, namely CIF (352×288). The tests are performed with the Coast Guard and Foreman sequences, both with the full 300 frames. The second approach, backward motion

estimation, uses the same conditions as for SVFME. The FVFME is compared, as mentioned above, with several alternative side information creation schemes, where MX represents extrapolation and ME stands for motion estimation with the original frame available at the decoder. The results show the comparison between the MX schemes.

The FVFME comparison with the extrapolation schemes presented in [31] and [32] is shown in Table 2.2. By observation the FVFME yields the best PSNR results for almost every sequence. For further information, please read [22].

Table 2.2 – PSNR comparison between different side information creation schemes

Type of approach		FVFME	[32] MX	[31] MX
Sequences tested	Carphone	30.9	28.45	29.03
	Stefan	26.3	23.73	22.88
	Foreman	32.3	31.57	30.66
	Coastguard	34.1	31.31	30.82
	Mother	41.9	-	-
	News	33.5	-	-
	Hall	37.4	-	-
	Silent	34.0	34.26	33.62

The SVFME solution along with the solutions classified as SBME, IBME and BA, are compared in Figure 2.16. Note that each blue cross, in the graphics, represents an average PSNR value of the side information for each of the approaches.

Following the results in Figure 2.16, it is possible to conclude that backward motion estimation is better than SVFME, in terms of side information PSNR, for both sequences. The main reason is associated with the algorithm used for the hole filling process; whereas the SVFME uses spatial hole filling, the backward motion estimation uses temporal hole filling. SBME shows that, in some situations, it is as good as the IBME, as shown for the Foreman sequence. Even though the use of bilateral filtering improves the PSNR as seen in Figure 2.16, the improvement is not significant, due to the inherent problems related with filtering and objective quality. The PSNR may be better but the subjective visual quality of the frame could be lower. For further information, please read [21].

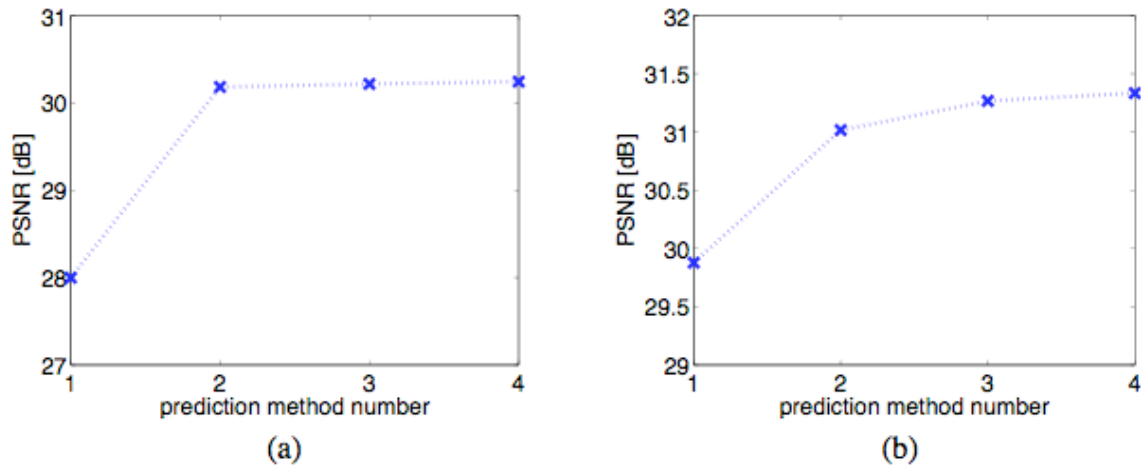


Figure 2.16 – Side information PSNR comparison for the a) Foreman and b) Coast Guard sequences: 1) SVFME; 2) SBME; 3) IBME; 4) BA.

2.2.4.5. Summary

Taking into account all the approaches presented above, the extrapolation methods present PSNR values similar to the ones in previous sections, but it is important to keep in mind that they only relate to the side information and not the full-decoded frame. The main advantage of the solutions described above is their ability to be used in low delay DVC systems, as the side information generation process does not introduce the latency common in interpolation framework schemes.

Chapter 3

Advanced Low Delay IST DVC Codec

This chapter intends to describe and evaluate the first low delay DVC codec developed by the author of this Thesis. This codec, named in the following Advanced Low Delay IST DVC (ALD-DVC) codec, is an evolution of the Low Delay IST DVC codec presented in the previous chapter and initially proposed in [24]. The ALD-DVC codec intends to improve some of the tools used in the Low Delay IST DVC codec, notably the side information creation process and the correlation noise modeling, and to remove the unrealistic assumptions still present in the Low Delay IST DVC codec proposed in [24] since original key frames are still used at the decoder. It is important to stress that the usage of original key frames makes the video codec unrealistic since this is not possible in practice; this means the Low Delay IST DVC codec [24] is not a practical solution and the performance results are biased by this limitation, notably because artificially good side information is used. This will not happen with the ALD-DVC codec to be proposed in this chapter, which is already a fully practical low delay DVC solution.

Since the ALD-DVC codec uses some similar tools to the Low Delay IST DVC codec, only the ones that are novel or improved will be presented in this chapter, as the others remain exactly the same, this means, as proposed in the Low Delay IST DVC codec [24].

3.1. Codec Architecture

This section presents the architecture and a brief walkthrough of the ALD-DVC codec developed by the author. This architecture is very similar to the architecture of the Low Delay IST DVC codec although the algorithms for the various modules are not the same; a major difference is the inclusion of the H.264/AVC Intra codec for the key frames which did not exist in the Low Delay IST DVC codec since originals were assumed to be available at the decoder.

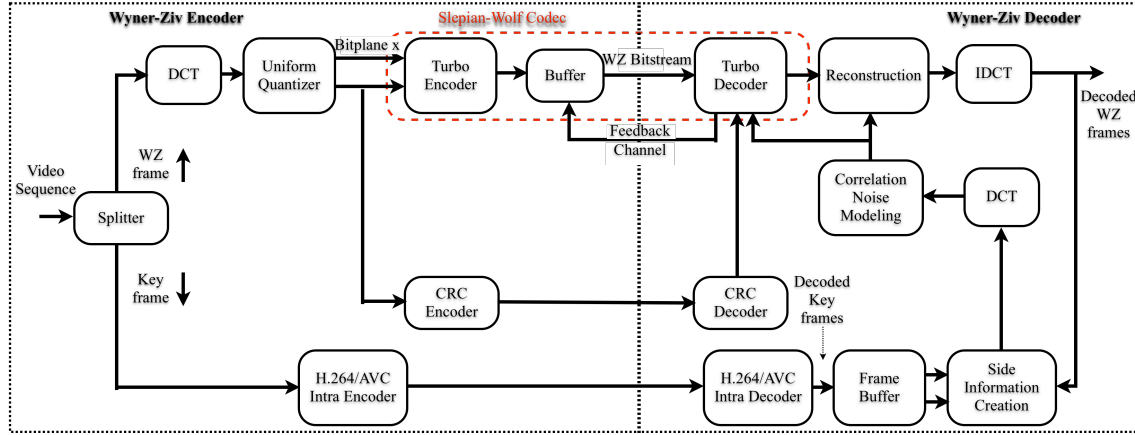


Figure 3.1 – ALD-DVC architecture.

As the ALD-DVC codec uses at its core the extrapolation-based Low Delay IST DVC architecture, most of the encoder and decoder modules remain the same. The ALD-DVC and the Low Delay IST DVC codecs are both based on the Stanford DVC approach and, thus, present an architecture similar to the DISCOVER DVC codec [12]. However, while the DISCOVER DVC codec adopts an interpolation-based side information creation process, the ALD-DVC codec adopts an extrapolation-based side information creation process as in the Low Delay IST DVC codec due to the low delay requirement adopted in this Thesis.

In the WZ encoding process, there are no novelties regarding the Low Delay IST DVC codec and also the DISCOVER DVC codec; thus, the encoding modules are exactly the same as those presented in Section 2.2.1. However, there is a difference in the frames encoding order since in the interpolation-based approach the next GOP key frame has to be encoded before the in-between WZ frames while in the extrapolation-based approach all frames are encoded in the acquisition order since there is no past and future interpolation. Naturally, this different order also applies to the decoding process. For example, if the GOP size is 4, this means one key frame and 3 WZ frames repeated in a regular GOP pattern, the encoding and decoding orders will be 1, 5, 3, 2, 4 for the interpolation-based approach as described in [20] and 1, 2, 3, 4, 5, 6, ... for the extrapolation-based approach as no future frames are needed to decode any current frame. Figure 3.2 shows the process explained above for GOP size 4.

A brief description of the encoding and decoding processes is given below.

Encoding Process:

- **Frame Classification** – The first step of the encoding process is to classify the video frames in the original sequence as WZ frames or key frames. Key frames are encoded using H.264/AVC Intra and WZ frames are encoded using a DVC approach. In the ALD-DVC codec, there is a need to have two initial, successive key frames (frames 1 and 2) since two successive frames are needed at the decoder for the Correlation noise modeling and Side Information Creation modules and it will be explained later. From this point on, depending on the GOP size, WZ and key frames appear in a regular temporal GOP pattern; for example, using GOP size 2, the odd frames are always WZ frames and the even frames are key frames, with the exception of the two initial frames. In the following, the encoding and

decoding processes for the WZ frames will be explained since the key frames are coded using the well known H.264/AVC Intra codec [37].

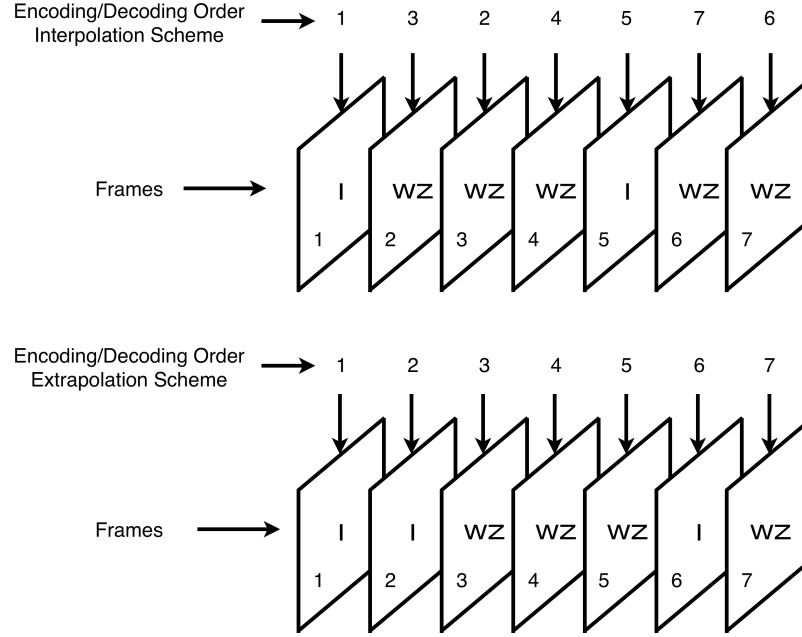


Figure 3.2 – Encoding/decoding orders for interpolation and extrapolation-based DVC schemes for GOP=4.

- **DCT** – A 4×4 block based DCT transform is applied to each WZ encoded video frame. The coefficients of the entire WZ frame are grouped together, according to their position inside the 4×4 blocks forming the DCT coefficient bands, i.e. given the 16 positions in the 4×4 block, 16 coefficient bands are formed.
- **Uniform Quantizer** –The uniform quantizer behaves differently for the DC band and the following AC bands. Regarding the DC band, as these coefficients express the average energy of the 4×4 block, they have typically high amplitude positive values and, thus, they are quantized using a scalar quantizer without a symmetric quantization interval around the zero amplitude [17].
- The AC bands are quantized using a uniform scalar quantizer with quantized bins evenly distributed around zero to reduce the block artifact effect. The zero bin has twice the bin size in order to accommodate more coefficients as they tend to concentrate around zero. This way the matching probability between the quantized bins of the WZ and side information (SI) frames increases, saving bitrate. In order to adapt the quantization step range to the AC bands, a dynamic approach is performed, increasing or decreasing the bin size to adjust the quantization step and, thus, increasing the coding efficiency. The step size W is given as $W = 2|V_k|_{\max} / (2^{M_k} - 1)$ where $|V_k|_{\max}$ represents the highest absolute value for band b_k , and 2^{M_k} the corresponding number of quantization levels used. After this quantization process, bitplane extraction is performed, grouping the quantized symbols bits of the same significance [18].

- **Turbo Encoder** – As the bitplanes generated in the Uniform Quantizer module arrive at the turbo encoder, the turbo encoding process begins, starting with the most significant bitplane array. The parity information generated is then stored in a buffer and sent to the Turbo Decoder upon request through the feedback channel.
- **CRC Encoding** – A CRC-8 (extra 8 bits) checksum is computed for each bitplane to be transmitted to the decoder; this information will help later the decoding process to achieve a vanishing error probability for each decoded bitplane.

Decoding Process:

- **Side Information Creation** – This is one of the most important modules in the decoder architecture as the availability of good side information reduces the number of rate requests made by the decoder through the feedback channel and, thus, the rate to achieve a certain decoded quality. This module can be divided into 4 sub-modules, each performing a certain task: motion estimation, motion field smoothening, motion projection and, finally, treatment of overlapping and uncovered areas [24]. In this case, the side information is generated using an extrapolation approach with two previously decoded frames, either key frames or WZ frames.
- **DCT** – As the ALD-DVC codec works in the transform domain, a DCT transform is applied to the generated side information frame. This follows the exact same procedure as the DCT module in the encoder.
- **Correlation Noise Modeling** – The residual statistics between the WZ DCT coefficients and the Side Information DCT coefficients is modeled by a Laplacian distribution which parameter has to be online estimated; the Laplacian parameter estimation may be performed at different granularity levels, notably band and coefficient levels [15].
- **Turbo Decoding** – As soon as the turbo decoder receives information about the residual statistics for a given bitplane of a DCT coefficients band b_k provided by the Correlation noise modeling module, the decoded quantized symbol stream associated to each DCT band b_k bitplane can be obtained through the turbo decoding procedure using successive chunks of parity bits, sent by the encoder. For each band b_k , the turbo decoder starts by decoding the most significant bitplane array, proceeding in an analogous way to the next bitplanes M_{k+1} in each band. The successful decoding of a bitplane is determined by a request stopping process described in the next bullet, which includes two steps: a request stopping criterion and a CRC-8 checking. In case the decoding fails, this means the request stopping process does not give a positive result, another request for more parity bits is sent to the encoder through the feedback channel. Once all the bitplanes have been successfully decoded for a band, the turbo decoder proceeds to the next band b_{k+1} . This process stops when all the DCT coefficients/bands have been decoded.
- **Request Stopping Process** – As explained above, the turbo decoder reaches a successful decoding for a bitplane, and thus stops decoding it, as soon as two conditions are verified. First, the target error probability for a given bitplane has to be achieved by reaching a certain value for the error probability computed based on the logarithmic likelihood ratio (LLR) threshold of 10^{-3} [12]. After, if this value is reached while decoding a certain bitplane, two situations might arise: i) if the CRC-8 is used and positively checks the decoded bitplane, then the very low residual error rate is confirmed and the decoding proceeds to the next bitplane; ii) if the received CRC-8 does not positively check the decoded bitplane, then more bits are asked from the encoder through the feedback channel and another run of turbo decoding is performed. To check the decoded bitplane, the CRC-8 (8 bits) checksum computed at the encoder and transmitted to the decoder for each decoded bitplane of each band is compared with the corresponding CRC-8

computed at the decoder based on the already decoded bitplane. The objective of this double process is to attain a vanishing error probability for each decoded bitplane in this case around 10^{-3} .

- **Symbol Assembling** – In order to obtain the bands b_k , the decoded bitplanes must be grouped together. After obtaining all the DCT coefficient bands for which WZ bits were transmitted, the side information DCT coefficient bands are used to complete those bands for which no WZ bits were transmitted due to quantization.
- **Reconstruction** – As soon as all the quantized symbols are obtained, the matrix with the decoded DCT coefficients for each 4×4 block is reconstructed using the same procedure as described in the previous chapter for the DISCOVER DVC codec [20].
- **IDCT** – Similar to the DCT module, an inverse 4×4 DCT transform is finally applied to all blocks in the entire frame obtaining the final decoded frame.
- **Frame Remixing** – In order to obtain a properly ordered video sequence, key frames and WZ frames must be sorted conveniently.

3.2. Extrapolation-based Side Information Creation

This section intends to describe in detail the extrapolation-based side information generation process used in the proposed ALD-DVC codec. This process considers the same modules as in the Low Delay IST DVC codec which architecture is shown in Figure 3.3. Since the description of the algorithms for the various modules is rather short in [24], the author of this Thesis had to reinvent most of them; moreover, there are also some novel ideas, notably for the treatment of the holes and overlapped pixels in the projected frame as well as for the correlation noise modeling process. A detailed architecture of the Side Information Creation module is presented in Figure 3.3.

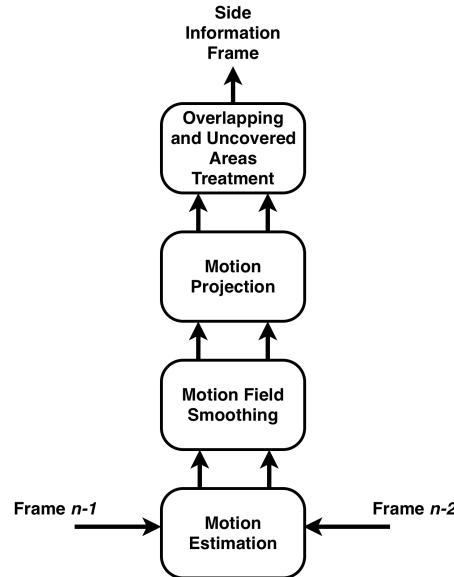


Figure 3.3 – Side Information Creation module.

The details of the four sub-modules in the Side Information Creation module are presented in the following sections.

3.2.1. Motion Estimation Sub-Module

The first module in the side information creation process is the motion estimation module, which has the objective to create a motion vector field, which will serve to project/extrapolate the side information for the next frame to decode. Given a current frame n , this module uses the two previously decoded frames, $n-1$ and $n-2$ (either key frames or WZ frames), with the purpose to build a motion vector field. The algorithm used in the creation of this motion vector field, commonly known as Full Search Block Matching (FSBM) is presented below:

- **Block Candidate Search** - Taking frame $n-1$, and using a specific block size, e.g. 8×8 samples, a search is performed in frame $n-2$ for each block in frame $n-1$ at position (x,y) , referring to the central position of each block, trying to find the best suitable match around that position (x,y) . As there is no gain in performing the search in the entire $n-2$ frame, a window surrounding position (x,y) is used to limit the complexity of the search; typically, the search window has a size of 16 samples in both directions. This block candidate search is performed for every block in the frame $n-1$.
- **Error Computation** - To find the best match in the previous $n-2$ frame, this search process uses an error metric known as Weighted Mean Absolute Matching Difference (WMAD) [30]. The WMAD is computed based on the absolute difference between the block in frame $n-1$ and the block being analyzed in frame $n-2$, as in equation (3.1), weighted by a factor depending on the distance between the two blocks being analyzed; in this way, the WMAD provides a better metric than the simple Mean Absolute Difference (MAD):

$$WMAD = \frac{\left| \sum_{x=0}^7 \sum_{y=0}^7 X_{n-1}(x,y) - X_{n-2}(x + d_x, y + d_y) \right|}{BlockArea} \times (1 + K \sqrt{d_x^2 + d_y^2}) \quad (3.1)$$

where X_{n-1} and X_{n-2} represent frames $n-1$ and $n-2$, respectively, and (x,y) represent the positions within the 8×8 block for which the absolute difference is being computed. *BlockArea* represents the entire block area in samples, in this case 64 assuming an 8×8 block size. The variable K represents the smoothness constant, a penalty that accounts for the furthest positions of the search range, with value 0.05 (experimental result). The variables dx and dy refer to the displacement between the blocks in frame $n-1$ and frame $n-2$ being compared, i.e. the distance between them in both directions, x and y . If the two blocks are collocated (they have the same position in the two frames), then the WMAD is nothing more than the MAD as $dx = dy = 0$; however, as $dx + dy$ increases, the impact on the mean absolute difference will be larger, increasing the WMAD. By performing the motion search with this WMAD criterion for all the blocks in frame $n-1$, a motion vector field is created.

3.2.2. Motion Field Smoothing Sub-Module

The second sub-module in the Side Information Creation module targets the smoothing of the motion vector field created in the previous step. Taking into account that the motion vector field is not always perfect and may show some rather random vectors, there is a need to add some robustness to the motion estimation process, thus creating a smoother and more reliable motion field as follows:

- **Neighbor Blocks Definition** - To address the objective above, for each specific block and the associated motion vector in frame $n-1$, a better and improved motion vector is obtained using also the motion vectors from the neighbor blocks.

Depending on the position of the block in the frame with a specific motion vector, more or less neighbor blocks will contribute to this process. For example, while the first block in the frame, this means the block in the top left corner, has only 3 adjacent blocks, a central block in the middle of the frame has 8 adjacent blocks.

- **Motion Vector Field Smoothing** – Having defined the neighbor blocks to be used, a new motion vector is computed as the median value, for both components (x,y) , of all the motion vectors of the available neighboring blocks and also the current block. Note that the motion vector for the current block under processing is always included in the median computation. This median value becomes the new motion vector for that specific block.

It is important to note that the use of the mean of the available motion vectors instead of the median would provide a less reliable solution as the average may be strongly conditioned by a single ‘very bad’ value which does not happen with the median which ‘filters’ the ‘outliers’.

3.2.3. Motion Projection Sub-Module

In order to create the extrapolated frame n , this means the side information for the next WZ frame to be decoded, a motion projection is performed for each block in frame $n-1$, in the next sub-module of the side information creation process. By applying the motion vector field resulting from the second sub-module to every 8×8 block in frame $n-1$, an extrapolated frame n is obtained, as seen in Figure 3.4. This process may lead to two types of problems addressed by the next sub-module: overlapping areas and holes corresponding to uncovered areas in the projected frame.

3.2.4. Overlapping and Uncovered Areas Treatment Sub-Module

The fourth sub-module addresses the problems resulting from the previous sub-module where a frame projection was performed with the previously computed motion field: overlapping and uncovered areas (holes) in the projected frame.

A) Overlapping Areas

The overlapping areas correspond to the areas in the projected frame, which are covered by more than one sample projected from the previous frame $n-1$; this means there is more than one estimation value for the same frame position and, thus, some unique solution must be determined.

- **Overlapping Areas Filling** - As proposed in [24], the solution adopted has been to average the alternative, competing values for the overlapping samples, and use those values as the final estimated value. As the estimation value must be an integer, the average above is truncated.

B) Uncovered Areas

The uncovered areas correspond to the areas in the projected frame, which are not covered by any sample projected from the previous $n-1$ frame; this means there is no estimation value for those frame positions. The solution adopted here to fill those uncovered areas has been to average (and truncate) the values of the surrounding projected samples.

- **Uncovered Areas Detection** - The scanning algorithm to find the uncovered areas within each frame is performed from top to bottom and from left to right. Special situations include the presence of a hole for the first sample (top-left) in the frame, which is solved by copying the sample in the same position from the previous frame.

- **Uncovered Areas Filling** - The detected uncovered areas are filled using the average value for the surrounding 8 pixels, excluding those that are also in uncovered areas. Again, as the estimated value has to be an integer, truncation of the average is performed. Figure 3.4 shows an example of the process described where the holes correspond to the black zones in the frame.

Considering the four test sequences which will be used later for performance evaluation, notably Hall Monitor, Coastguard, Soccer and Foreman, one frame was chosen to exemplify the improvements obtained by using the proposed motion field smoothing module and the overlapping and uncovered areas treatment process; the results are shown in Figure 3.4. First, column a) represents the side information frame without using the proposed motion vector field smoothing process; next, column b) shows, for the same frame, the improvements brought with the usage of the motion field smoothing process. Finally, column c) demonstrates the application of the last sub-module, the overlapping and uncovered areas treatment, where the holes are represented by the black zones which are eliminated with the proposed algorithms.

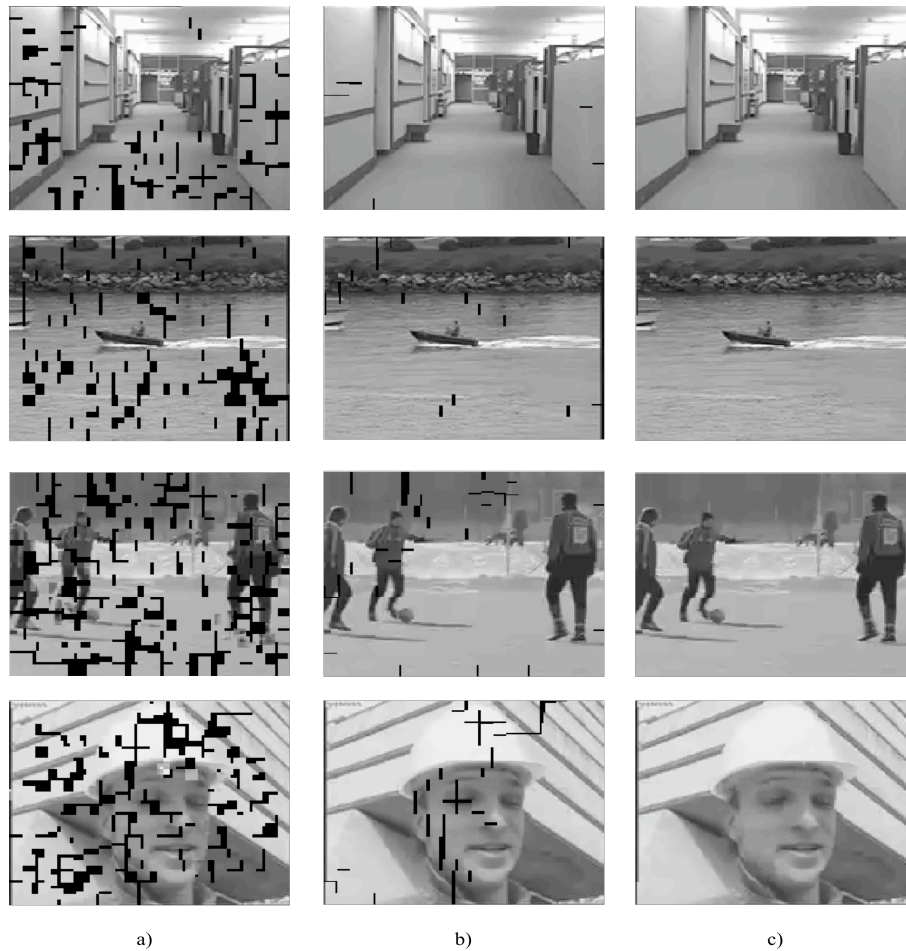


Figure 3.4 – a) Side information after motion projection without motion vector field smoothing; b) Side information after motion projection with motion vector field smoothing; c) Side information after motion projection with motion vector field smoothing and treatment of overlapping zones and holes.

3.3. Correlation Noise Modeling

The correlation noise modeling (CNM) is a very important module in the codec architecture with the main objective to provide the turbo decoder with a measure of confidence on the generated side information, and also help in the reconstruction process. In this context, the correlation noise refers to the difference between each original (quantized) WZ frame available at the encoder and the corresponding side information created at the decoder. In this context, it is important to distinguish two main cases in terms of correlation noise modeling: offline correlation noise modeling which stands for the process where the CNM parameters are obtained assuming that both the original data and the side information are simultaneously known (which is impossible in a practical DVC setup) and, on the contrary, online correlation noise modeling which corresponds to the process where the CNM parameters are estimated at the decoder in a practical way, thus without using any original data [15].

Many of the available DVC codecs use an offline modeling process, thus assuming that the original sequences are known at the decoder or the side information is available at the encoder, making it impractical and not very real world driven. As the objective of the ALD-DVC codec is to create an efficient but also practical DVC codec, the use of online correlation noise modeling is essential. In order to use online correlation noise modeling at the decoder, it is necessary to model the correlation noise using an adequate distribution to determine a measure of confidence on the generated side information to be used in the turbo decoding process.

In traditional video coding, the Laplacian distribution is typically used to model the distribution of the motion-compensated residual DCT coefficients. More accurate models can be found in literature, such as the generalized Gaussian distribution; however, the Laplacian distribution constitutes a good tradeoff between model accuracy and complexity and, therefore, it is often chosen. For the same reason, the Laplacian distribution is widely used to model the correlation noise in the DVC literature [15], and, therefore, it will be also adopted in this Thesis. This Laplacian distribution is typically characterized by a single parameter, which has to be estimated, the so-called α parameter.

This proposed correlation noise modeling process can be broken down into six steps as shown in Figure 3.5.

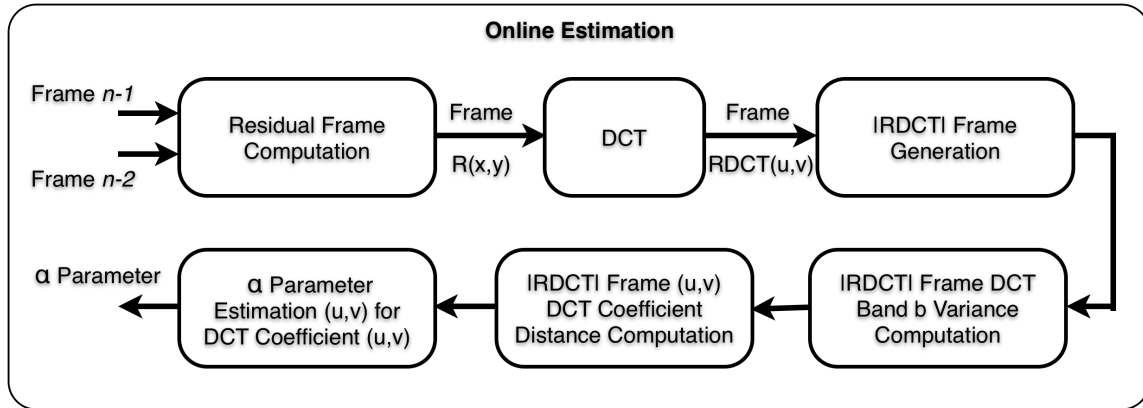


Figure 3.5 – Correlation noise modeling architecture.

In order to estimate the CNM α parameter, a residue R has to be computed at the decoder which should approximate (estimate) as much as possible the real correlation noise, this means the difference between the original (quantized) WZ frame available at the encoder and the corresponding side information created at the decoder. In this Thesis, this residue will

be computed using the frames involved in the extrapolation process, i.e. the frames used to define the motion field and project the side information for the current WZ frame. As the ALD-DVC codec performs in the transform domain, the correlation noise is nothing more than the residual between the DCT bands of the quantized WZ frame and the corresponding SI frame. It is important to note that the more accurate the correlation model, the higher will be the compression efficiency as less parity bits will be required for the turbo decoder to correct the same amount of errors in the side information frame. While indicating to the turbo decoder a low confidence than available on the side information will lead to expending more rate than needed, indicating a higher confidence will lead to decoding errors with an intense negative subjective impact regarding the decoded video.

As mentioned above, a Laplacian distribution is adopted, as widely done in the literature, to model the distribution of the motion compensated residual DCT coefficients. The Laplacian distribution represents a good tradeoff between model accuracy and complexity, delivering good soft input information, i.e. conditional bit probabilities, to the turbo decoding process. This soft-input information is a measure of the confidence on the side information given to the turbo decoder, which should express the level of similarity between the WZ and SI DCT coefficients; the higher the similarity, the higher the confidence level given to the turbo decoder and the more efficient (less rate) will be the decoding process. A detailed description of the steps proposed for the Correlation Noise Modeling process is provided below; this process is an adaptation of the solution proposed in [15] for an interpolation-based DVC codec.

3.3.1. Residual Frame Computation

The first step of the CNM process begins with the computation of the residual frame which should estimate the difference between the original (quantized) WZ frame available at the encoder and the corresponding side information created at the decoder. There are three possible alternatives to compute the residue, depending on the situation, as the projected blocks in frame n clearly cannot have the same treatment as the overlapped and uncovered areas. Thus, this residue computation can be divided into 4 steps, being the first the identification of each type of situation, i.e. if it is a projected area, an overlapping area or an uncovered area. The remaining 3 steps are the residue computation for the projected areas, the residue computation for the overlapping areas and, finally, the residue computation for the uncovered areas.

Initially, the first draft for the extrapolated frame resembles the situation in columns a) or b) in Figure 3.4, a result from the motion projection of the blocks in frame $n-1$ to frame n , as seen in Figure 3.6.

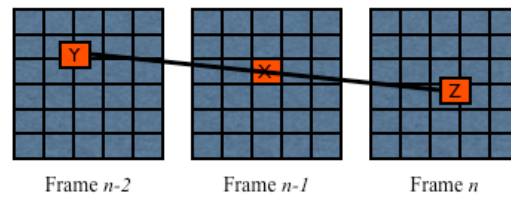


Figure 3.6 – Motion projection process.

- **Identification of Projected Areas versus Overlapping/Uncovered Areas** – There are areas in the current extrapolated frame n that were not obtained through the motion projection step, i.e. the values of some samples were obtained using the Overlapped and Holes Treatment sub-module presented before. In this context, it is necessary to

identify the areas that were obtained through motion projection in order to estimate the residue for those. All the other areas have a different residue calculation process detailed below.

- **Residue Computation for Projected Areas** - The residue calculation for the projected areas is no more than the subtraction between block X and Y that originated Z as seen in equation (3.2) assuming linear motion; if this residue is low, than this should mean that the motion is well modeled and there is high confidence on the generated side information. Given a block X in frame $n-1$, when performing the first step of the side information generation process, another block Y is found in frame $n-2$ that minimizes the WMAD, identifying a motion vector for each block in frame $n-1$. By applying motion projection to block X , a block Z is obtained in frame n , the extrapolated frame or side information. In this context, the residue is computed as:

$$R(x,y) = X_{n-1}(x - d_{x1}, y - d_{y1}) - X_{n-2}(x - d_{x2}, y - d_{y2}) \quad (3.2)$$

where X_{n-1} and X_{n-2} represent the two previously decoded frames used in the side information creation process, (dx_1, dy_1) represents the motion vector that created the final block Z , i.e. the motion vector after the motion field smoothing. The motion vectors (dx_2, dy_2) have twice the size of the motion vectors (dx_1, dy_1) , since frame $n-2$ is being linearly projected to a time instant, two frame periods away. The resulting R frame at each position (x,y) corresponds to the difference between the projection of the two past frames used for the motion field computation. It is important to note that the residue computation used here concerns only the positions in the current frame n for which the blocks were obtained through motion projection.

The residual image computed above has two problems inherent to the side information generation process, the holes and overlapping issues. For these areas, the residue computation proceeds as follows:

- **Residue Computation for Overlapping Areas** - If some of the motion projected blocks overlap when creating the SI frame, then according to the Residue Computation for Projected Areas step presented before, there is one residue value for each possible sample value available for that position. In this situation, it is proposed here to average and truncate the two computed residues.
- **Residue Computation for Uncovered Areas** – The holes are filled with the truncated average of the surrounding residue values, excluding those that have no value associated. Special situations include the presence of a hole for the first sample in the frame where no residue was computed and thus no averaging using neighbors is possible; in this case, a zero value is attributed to that residue.

Another solution was tried in terms of residue computation, which created first two projected frames: the first uses frame $n-2$ and the corresponding motion vectors with double size and results from applying the motion projection and the treatment of the holes and overlapping areas; the second frame is the extrapolated frame. Finally, the residue computation corresponds to the difference between these two projected frames. As this solution it yielded worse RD performance results, the first solution was adopted.

3.3.2. RDCT Computation and |RDCT| Frame Generation

The residue frame R obtained in the previous step has to be brought to the transform domain, as the information received in the turbo decoder regards the DCT bands from the encoded WZ frame.

- **RDCT Computation** - A 4×4 block based DCT transform is applied to the residual frame $R(x,y)$ in order to obtain the DCT coefficients frame $RDCT(u,v)$.
- **|RDCT| Frame Generation** – After, the absolute value for the frame $RDCT(u,v)$ is computed resulting in a $|RDCT(u,v)|$ frame.

3.3.3. RDCT Band b Variance Computation

The objective of this variance computation is to provide a reference value when classifying the DCT coefficients resulting from the Side Information Creation module as more or less reliable; this will also be used later in the computation of the α parameter, as proposed in equation (3.7).

- **Variance Computation** - Using equation (3.3), the RDCT band b variance $\hat{\sigma}_b^2$ is computed as in equation (3.4)

$$\hat{\sigma}_b^2 = E_b \left[(|RDCT|_b)^2 \right] - \left(E_b (|RDCT|_b) \right)^2 \quad (3.3)$$

where E_b represents the expected value, i.e the mean.

$$E_b (|RDCT|_b) = \frac{1}{J} \sum_{j=1}^J (|RDCT|_b)(j) \quad E_b \left[(|RDCT|_b)^2 \right] = \frac{1}{J} \sum_{j=1}^J \left[(|RDCT|_b)(j) \right]^2 \quad (3.4)$$

It is important to note that J is the DCT band size, this means the number of coefficients in each band, which corresponds to the ratio between the frame size in samples and the number of DCT coefficient bands (16 in this case).

3.3.4. α Parameter Estimation at DCT Band b Level

If the CNM is performed at band level, the α parameter is estimated using equation (3.5), requiring only the value of the variance computed in the previous step. If a more granular, and thus more accurate, correlation noise modeling is desired, than the α parameter has to be estimated at a lower level, e.g. at coefficient level, as proposed below.

$$\hat{\alpha}_b = \sqrt{\frac{2}{\hat{\sigma}_b^2}} \quad (3.5)$$

3.3.5. |RDCT| (u,v) DCT Coefficient Distance Computation

In order to distinguish more and less reliable DCT coefficient estimations through the side information generation process, a distance $D_b(u,v)$ computed as in equation (3.6), between the $RDCT(u,v)$ coefficient for a given band b and the $|RDCT|$ frame band average $\hat{\mu}_b$ is required.

$$D_b(u,v) = |RDCT|_b - \hat{\mu}_b \quad (3.6)$$

3.3.6. α Parameter Estimation at DCT Coefficient (u,v) Level

This final step is the estimation of the α parameter for the DCT coefficient at position (u,v) using equation (3.7).

$$\hat{\alpha}_b(u,v) = \begin{cases} \hat{\alpha}_b, & [D_b(u,v)]^2 \leq \hat{\sigma}_b^2 \\ \sqrt{\frac{2}{[D_b(u,v)]^2}}, & [D_b(u,v)]^2 > \hat{\sigma}_b^2 \end{cases} \quad (3.7)$$

The solution in (3.7), distinguishes two situations has proposed in [15]:

- In the **first** situation, the squared computed distance $[D_b(u,v)]^2$ obtained in the previous step is inferior or equal to the variance $\hat{\sigma}_b^2$ meaning that the block was well extrapolated; in this context, $\hat{\alpha}_b(u,v)$, computed as in equation (3.5), is used as the estimation for the α parameter for the (u,v) DCT coefficient.
- The **second** situation arises when the squared computed distance $[D_b(u,v)]^2$ obtained in the previous step is larger than the variance $\hat{\sigma}_b^2$. In this case, the projected frame generated in the Side Information Creation module is not accurate; hence, the alternative proposed, which provides the best results, is to use the squared computed distance $[D_b(u,v)]^2$ instead of the $\hat{\sigma}_b^2$ in equation (3.5) for obtaining the best α parameter estimation.

In summary, if the block is well extrapolated, the α parameter is estimated as the band variance indicating to the turbo decoder a high confidence on the side information block. If this is not the case, then the squared computed distance $[D_b(u,v)]^2$ is used, lowering the confidence of the turbo decoder on the side information generated block.

3.4. Performance Evaluation

This section intends to evaluate the RD performance of the ALD-DVC codec in comparison with relevant alternative codecs, notably DVC codecs and relevant standard-based video codecs. In order to obtain results capable of being compared with alternative codecs, the test conditions recommended by the DISCOVER project [12] and largely used in the international DVC literature were adopted, notably:

- **Test Sequences** – The four sequences used are Coastguard, Foreman, Hall Monitor and Soccer since they represent different types of video content.
- **Temporal and Spatial Resolution** – Sequences were coded at 15Hz with QCIF resolution (176×144 luminance samples).
- **Frames for each Sequence** – For each sequence, the entire length was used, namely 150 frames for the sequences Coastguard, Foreman and Soccer; for the Hall Monitor sequence, 165 frames (full length) were used.
- **GOP Sizes** – The GOP size may be 2, 4 or even 8; in case of omission, a GOP size of 2 is assumed.
- **Rate Distortion Points** – To define several RD trade-off points, eight quantization matrices were adopted as in [20]; the eight 4×4 quantization matrices define the quantization steps associated to the various DCT coefficients bands, see Figure 3.7. The use of quantization matrices from a) to h) in Figure 3.7 corresponds to a quality improvement but also to a bitrate increase. It is important to note that the key frames are encoded in such a way (this means using a quantization step) to have a similar average quality to the WZ frames in order the overall video quality does not have

significant temporal variations. It is important to notice that this claim regards the interpolation-based DISCOVER DVC codec [12].

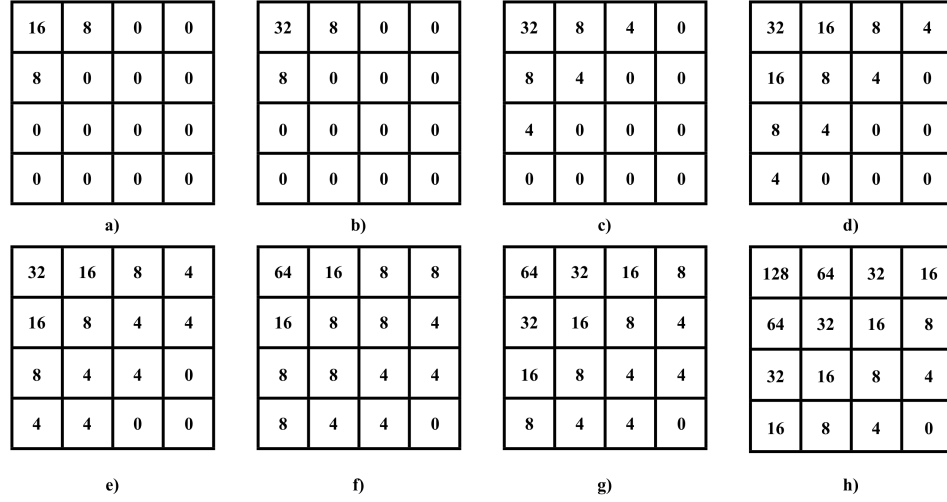


Figure 3.7 – Eight quantization matrices corresponding to the tested RD points.

- **Bitrate and PSNR** – As usual in the DVC literature, only the luminance of the original sequences is encoded and decoded and, therefore, used in the bitrate and PSNR computations. It is important to note that both the WZ and key frames bitrate and PSNR are accounted for as opposed to some DVC codecs presented in the previous chapter.

To allow a good knowledge on the ALD-DVC RD performance, the codec performance is compared with some of the state-of-the-art coding solutions, both DVC and standard based codecs; for the comparisons to be minimally fair, the selected standard based codecs do not use motion estimation at the encoder to guarantee that all have a rather similar, low encoding complexity. In this context, the benchmark video codecs selected are:

- **H.263+ Intra** – An important, although not anymore a state-of-the-art, benchmark as it is used in many DVC papers to check the RD performance results; no temporal redundancy is exploited in this codec.
- **H.264/AVC Intra** – The state-of-the-art on standard intra coding, in this case using the Main profile; again, it does not exploit temporal redundancy.
- **H.264/AVC Zero Motion** – As opposed to the H.264/AVC Intra benchmark, there is exploitation of the temporal redundancy in this codec, although without using motion estimation to limit the encoder complexity.
- **DISCOVER DVC** – Considered as the state-of-the-art in DVC codecs [12], the DISCOVER DVC presents itself as the best DVC benchmarking although using an interpolation approach to generate the side information; as such, its performance may only be taken as a limit to reach by the extrapolation-based DVC codecs.
- **IST DVC Interpolation** – This is another alternative DVC codec, largely corresponding to the interpolation-based version of the extrapolation-based Low Delay IST codec [35]. This comparison allows evaluating the impact of using an extrapolation-based scheme as opposed to an interpolation-based side information creation scheme. In practice, the Interpolation-based IST DVC codec is an evolution of the DISCOVER DVC codec.

3.5. Results and Analysis

3.5.1. RD Performance for GOP Size 2

The RD performance results for the ALD-DVC coded and the selected benchmarks are presented in this section.

As presented in Table 3.1, there are eight entries for each tested sequence, each corresponding to one of the eight quantization matrices adopted. Figure 3.8, Figure 3.9, Figure 3.10 and Figure 3.11 present the RD performance charts for the four tested video sequences in comparison with the predefined benchmarking alternatives.

Table 3.1 - Performance results for the ALD-DVC codec.

Coastguard		Foreman		Hall Monitor		Soccer	
Rate (kbps)	PSNR (dB)	Rate (kbps)	PSNR (dB)	Rate (kbps)	PSNR (dB)	Rate (kbps)	PSNR (dB)
86.04	27.58	79.94	27.65	86.85	30.77	62.88	27.45
106.14	28.23	96.74	28.3	97.68	31.34	78.59	28.03
111.61	28.37	109.21	28.85	101.5	31.43	91.7	28.92
172.16	30.23	179.66	31.36	139.6	33.81	155.89	31.55
191.25	30.73	189.24	31.47	143.38	33.84	163.22	31.63
256.17	32	248.05	32.73	180.58	35.42	216.82	32.74
309.15	33.01	331.85	34.86	211.37	36.89	294.24	34.7
514.77	36.16	531.83	38.34	321.75	40.29	521.92	38.41

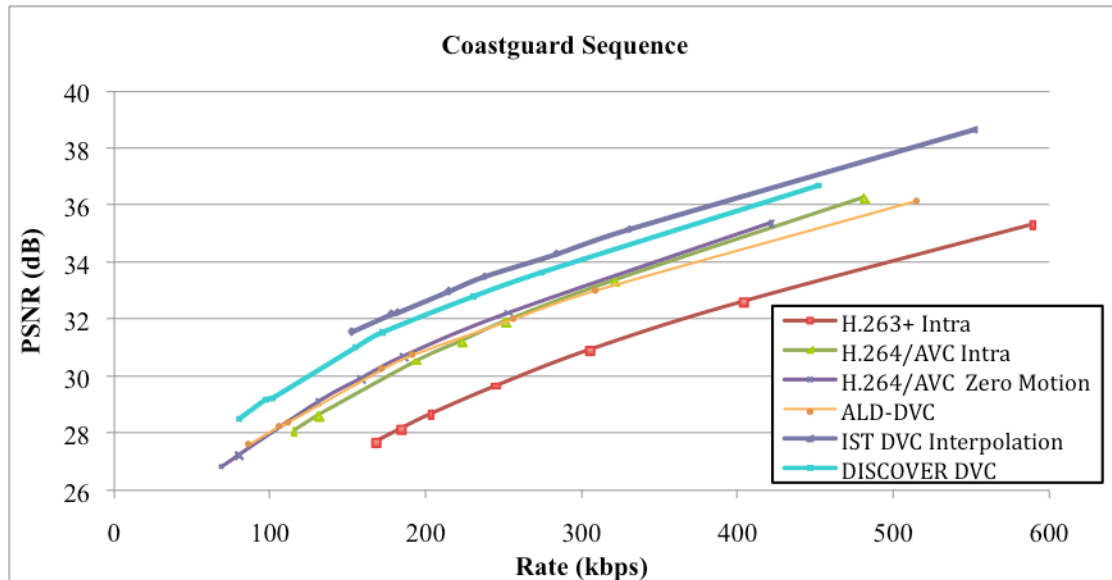


Figure 3.8 – RD performance comparison for the Coastguard sequence using GOP size 2.

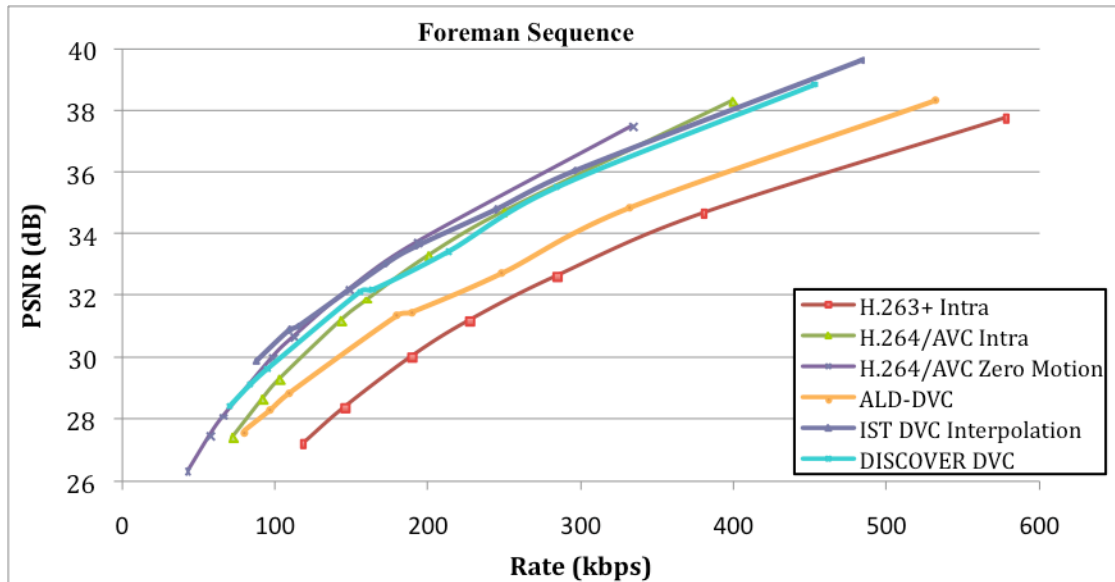


Figure 3.9 – RD performance comparison for the Foreman sequence using GOP size 2.

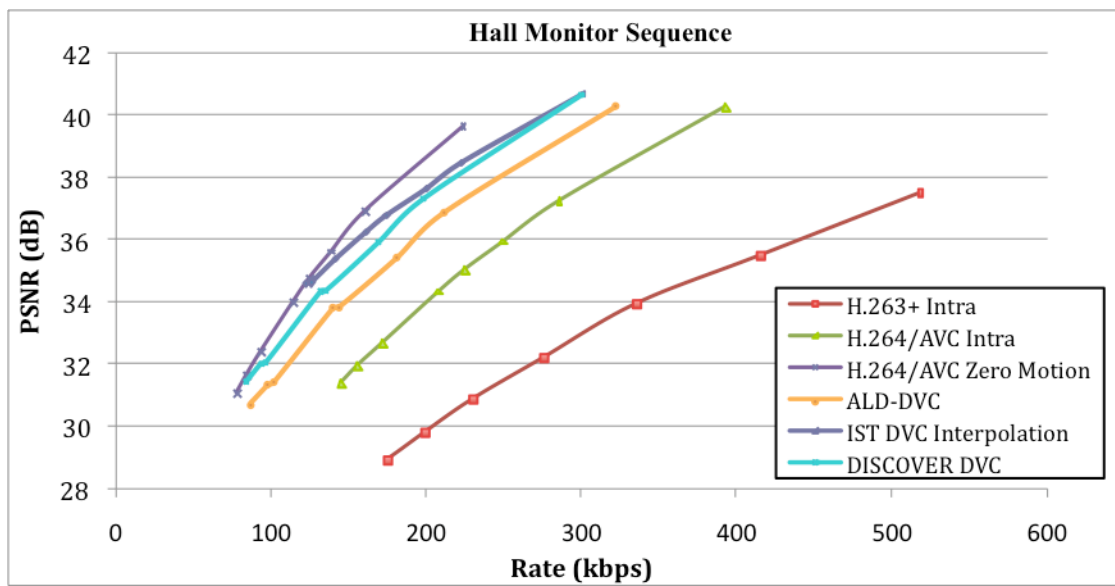


Figure 3.10 – RD performance comparison for the Hall Monitor sequence using GOP size 2.

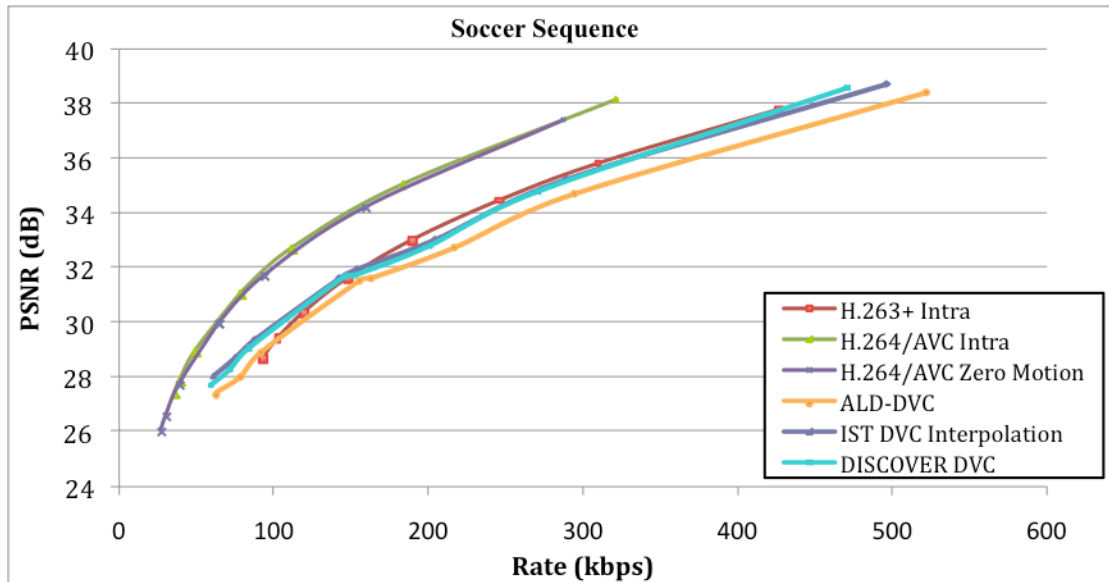


Figure 3.11 – *d* RD performance comparison for the Soccer sequence using GOP size 2.

- **ALD-DVC versus IST DVC Interpolation** – As expected, the IST DVC Interpolation codec performs better than the ALD-DVC codec for all the sequences tested, showing an increasing or decreasing gap between the two RD performances as the rate increases, depending on the video content. This RD performance gain is expectable as the IST DVC Interpolation codec takes benefit of a future frame (and associated delay), which does not happen with the ALD-DVC codec. The Coastguard sequence, with a very regular movement, shows a gap between these two codecs, which is approximately constant for the various rates, although increasing for the highest rates. Regarding the Foreman and Soccer sequences, those with more complex motion, the RD performance gap widens with the increase of rate/quality. For the Hall Monitor sequence, characterized by small motion and fixed background, the gap is rather constant, eventually closing for the highest rates.
- **ALD-DVC versus DISCOVER DVC Codec** – As said before, the IST DVC Interpolation scheme is an evolution of the DISCOVER DVC codec; thus, it presents better RD distortion curves for most sequences, namely the Coastguard, Foreman and Hall Monitor sequences (and very similar for Soccer). Consequently, the gap between the DISCOVER DVC codec and the ALD-DVC codec is smaller than the gap between the IST DVC Interpolation codec with the exception of the Soccer sequence (this is due to the complex motion in that particular sequence).
- **ALD-DVC versus standard Intra codecs** – Regarding the comparison between the ALD-DVC codec and standard Intra codecs, there are good perspectives. With the exception of the Soccer sequence, which does not correspond to the most relevant DVC application scenario, the behavior for the other three tested sequences shows that the ALD-DVC codec performs much better than the H.263+ Intra codec. From these three sequences, the Foreman sequence is the one where the ALD-DVC codec is closer to the H.263+ Intra codec, presenting a gain up to 1-1.2 dB. The Coastguard sequence also shows good improvements, notably from 1.5 dB to 2 dB, but the Hall Monitor sequence overcomes all of them by showing RD gains up to 6 dB regarding the H.263+ Intra codec, mainly due to the static background. This indicates that the ALD-DVC codec is an efficient solution for video content with low and well behaved motion. Regarding the Soccer sequence, the ALD-DVC codec performs slightly worse than H.263+ Intra, notably due to the

random and high motion. When it comes to the state-of-the-art H.264/AVC Intra codec, the most important benchmarking, the ALD-DVC codec only overcomes it for the Hall Monitor sequence, with gains up to 2 dB. For the Coastguard sequence, the performance is approximately the same. For the other sequences, the ALD-DVC coded is outperformed by the H.264/AVC Intra codec showing losses up to 3 dB for the Soccer sequence and up to 2 dB for the Foreman sequence.

- **ALD-DVC versus standard Inter Zero Motion codecs** – As the Inter Zero Motion codecs already explore some of the temporal redundancy, they are more difficult to overcome in terms of RD performance; notably for sequences with little motion, its compression behavior is very good since temporal redundancy is fully exploited for the static zones. For the Hall Monitor sequence, the ALD-DVC codec is surpassed by the H.264/AVC Zero Motion, which shows gains up to 2 dB. For the Coastguard sequence, these two codecs show a RD curve at a similar level, with little or no difference at all. Regarding the Foreman sequence, the two curves appear to start at the same level but the gap widens with the increase of rate/quality, i.e. the use of ‘better’ quantization matrices; for the worst case scenario, the ALD-DVC codec shows a loss of up to 2 dB. Finally, the Soccer sequence registers up to 3 dB losses for the ALD-DVC codec, due to the difficulties with the intense and varied motion, very common in this sequence.

As it can be seen from the RD performances for the four tested sequences, the developed DVC codec still has room for improvement as its behavior is still below some of the adopted benchmarks. The best performance happens for the Hall Monitor sequence, where the ALD-DVC codec is well above the Intra coding solutions; this is the exception in the four tested sequences, as this one is characterized by almost no motion during its entire duration. This is an important exception since Hall Monitor corresponds precisely to the type of applications, which DVC technology seems to fit better. Moreover, it is important to remind that in these experiments both the rate and PSNR are computed for the full set of coded frames, as opposed to many of the codecs presented in the previous chapter, namely [22] and [24].

3.5.2. RD Performance for Longer GOP Sizes

All the tests above were performed using GOP size 2; however, it is important to analyze the behavior of the ALD-DVC when using larger GOP sizes, such as 4 and 8, for all the four sequences. The results for larger GOP sizes are presented in Figures 3.12, 3.13, 3.14 and 3.15.

- **ALD-DVC GOP Size Variation** – As shown in Figures 3.12, 3.13, 3.14 and 3.15, if the GOP size increases, the RD performance tends to degrade as shown in the charts for the Coastguard, Foreman and Soccer sequences. This behavior is largely motivated by the fact that in the ALD-DVC codec the usage of the same quantization steps as in the DISCOVER test conditions lead to key frames with much better quality than the WZ frames (see Figures 3.16, 3.17, 3.18 and 3.19). In this context, the more key frames are used, the better is the side information quality and thus the overall RD performance. Moreover, when less key frames are used, the decoded WZ SI frames show inferior quality and then the motion estimation process becomes less reliable. The Hall Monitor sequence is the exception, as the motion complexity is rather low, and thus good motion estimations are easier to perform, thus presenting a better RD distortion curve for the larger GOP sizes. If the key frames were coded with a similar quality as the extrapolated frames, then the behavior described should largely disappear would be largely reduced.

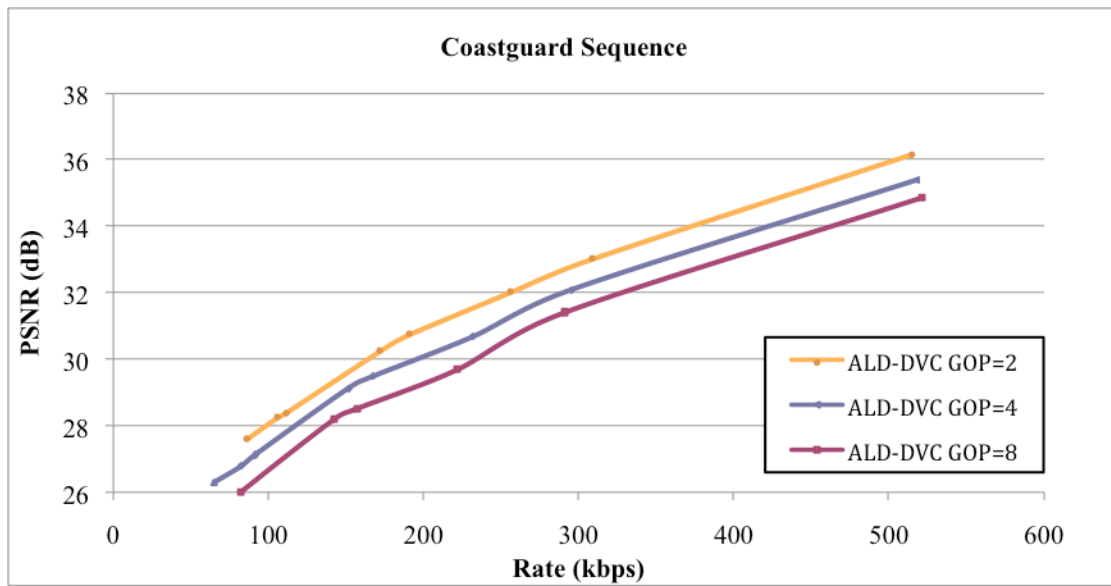


Figure 3.12 – ALD-DVC RD performance comparison for the Coastguard sequence, using GOP size 2, 4 and 8.

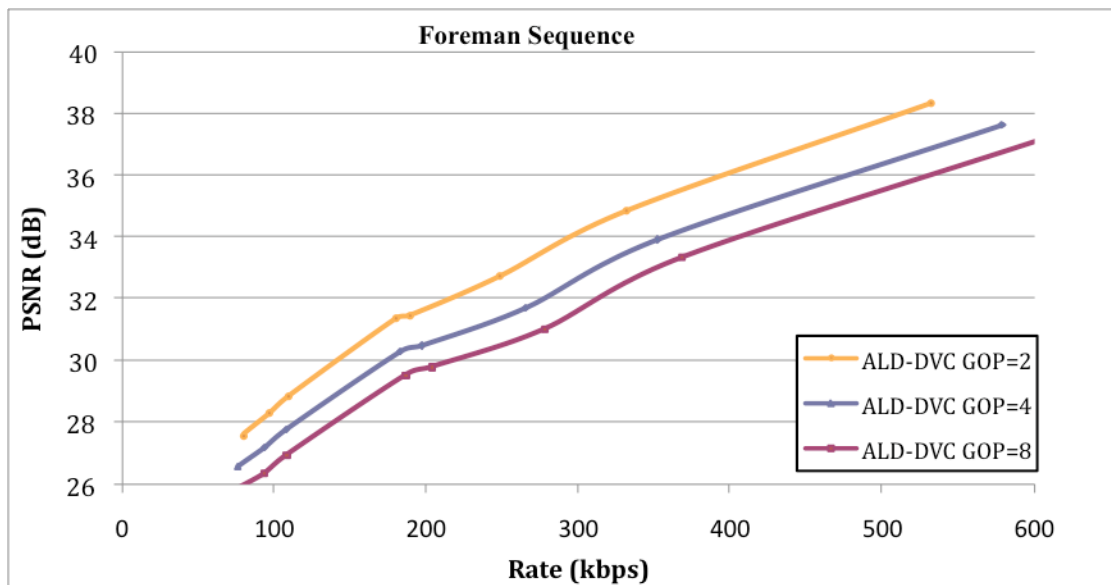


Figure 3.13 – ALD-DVC RD performance comparison for the Foreman, sequence using GOP size 2, 4 and 8.

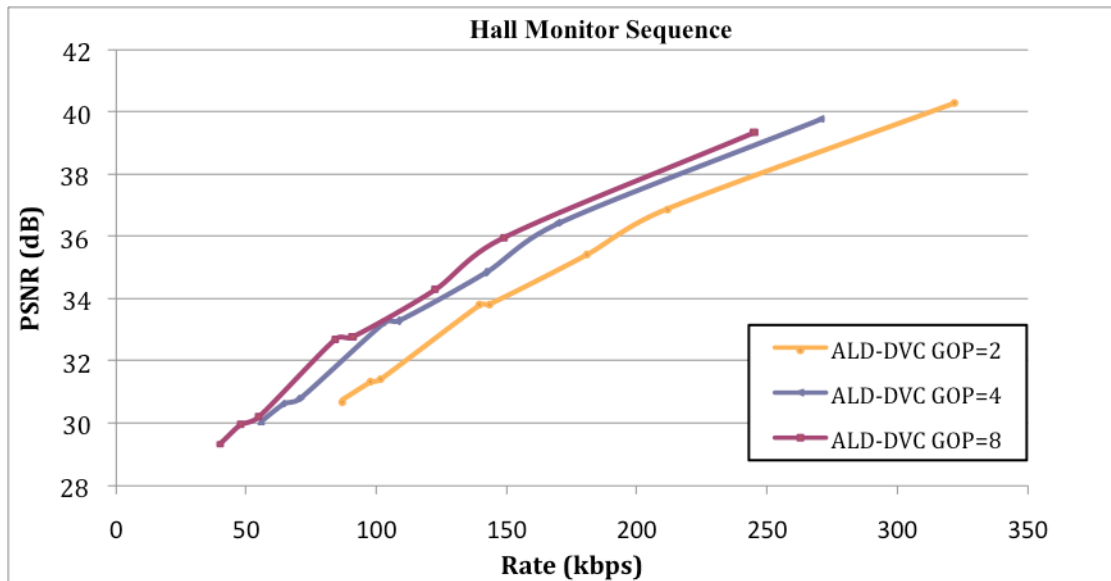


Figure 3.14 – ALD-DVC RD performance comparison for the Hall Monitor sequence, using GOP size 2, 4 and 8.

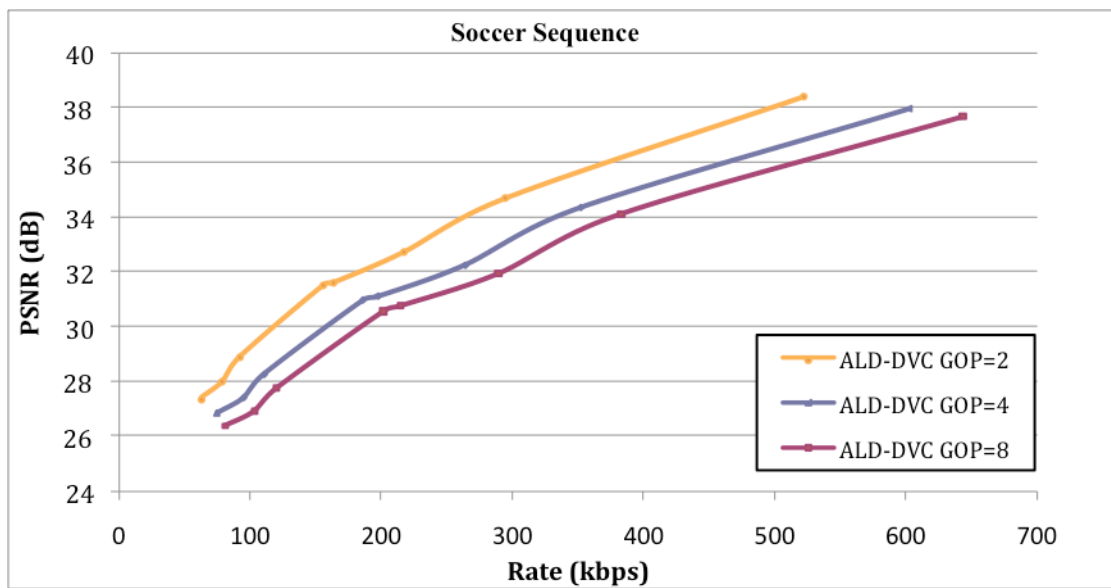


Figure 3.15 – ALD-DVC RD performance comparison for the Soccer sequence, using GOP size 2, 4 and 8.

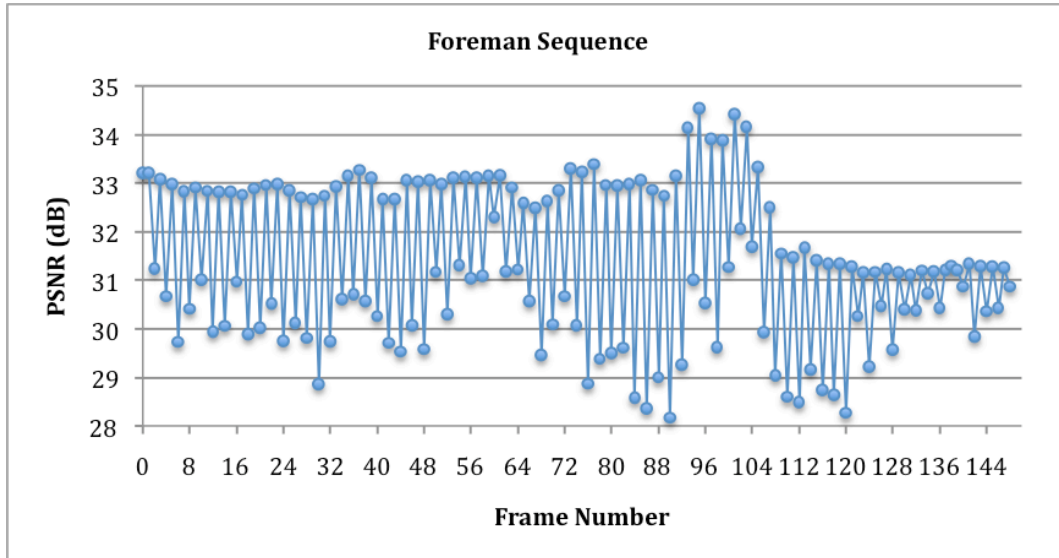


Figure 3.16 – PSNR versus Frame Number for Foreman sequence 15 Hz using GOP size 2 and $Q_i = 4$.

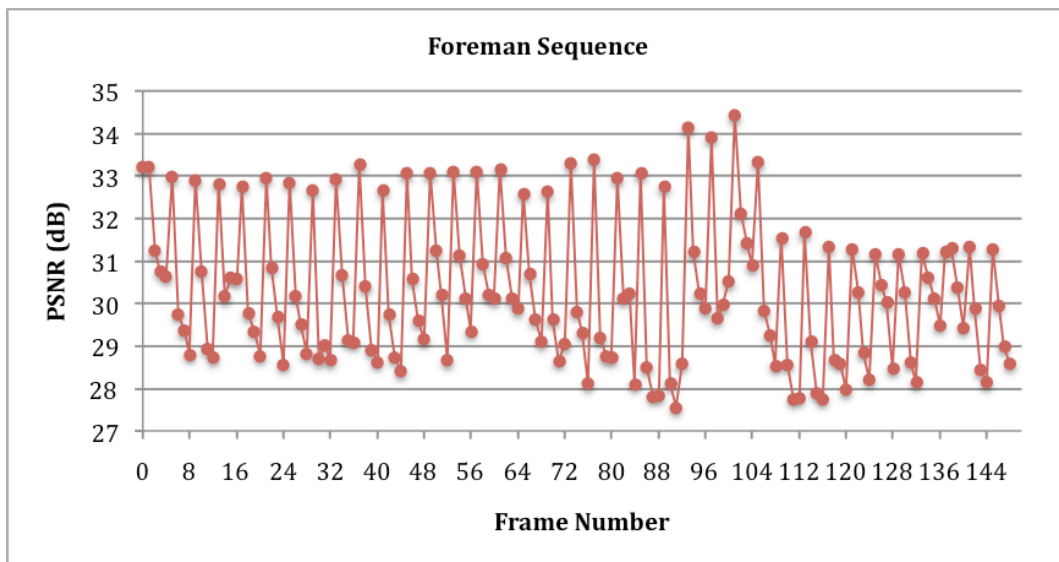


Figure 3.17 – PSNR versus Frame Number for Foreman sequence 15 Hz using GOP size 4 and $Q_i = 4$.

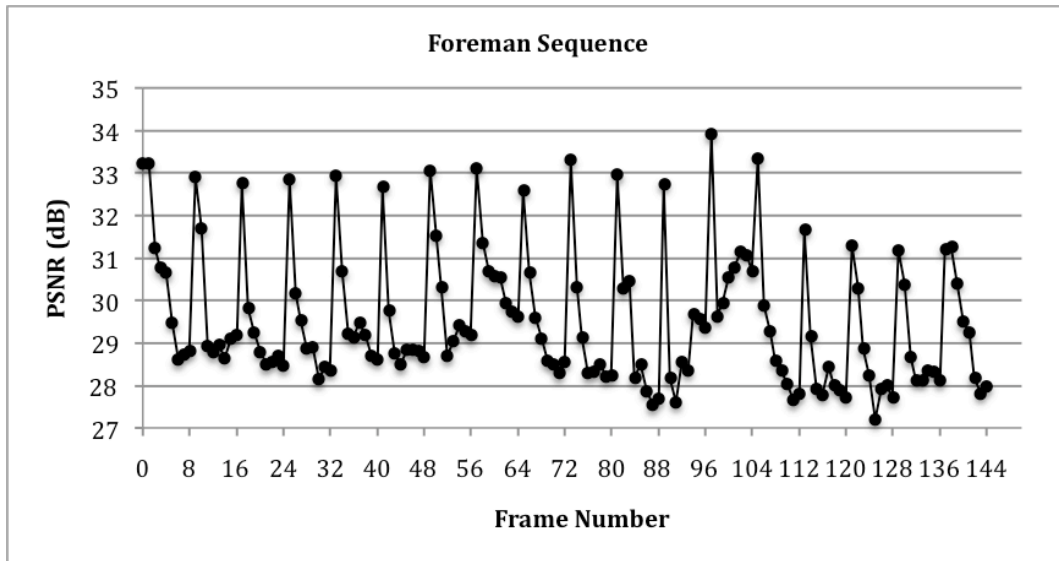


Figure 3.18 – PSNR versus Frame Number for Foreman sequence 15 Hz using GOP size 8 and $Q_i = 4$.

Figures 3.16, 3.17, 3.18 and 3.19 show the temporal evolution of the PSNR for the Foreman sequence coded with different GOP sizes and $Q_i = 4$; from these charts, it is clear that the key frames are currently coded with a much better average quality than the WZ frames. In fact, it is possible to distinguish two quality levels in Figures 3.16, 3.17 and 3.18 corresponding to the quality of the key frames and to the quality of the WZ frames; sometimes, this quality gap goes up to 3 dB. Hence, by reducing the number of key frames, the motion estimation gets less reliable as stated above, and the WZ frames present lower quality. It is important to note that although the PSNR temporal evolution is only presented the Foreman sequence with GOP size 2, 4 and 8, and $Q_i = 4$, this behavior is consistently the same for all the other sequences tested. Figure 3.19 shows the overlapping of part of the charts in Figures 3.16, 3.17 and 3.18, till frame 48, showing clearly the behavior mentioned above.

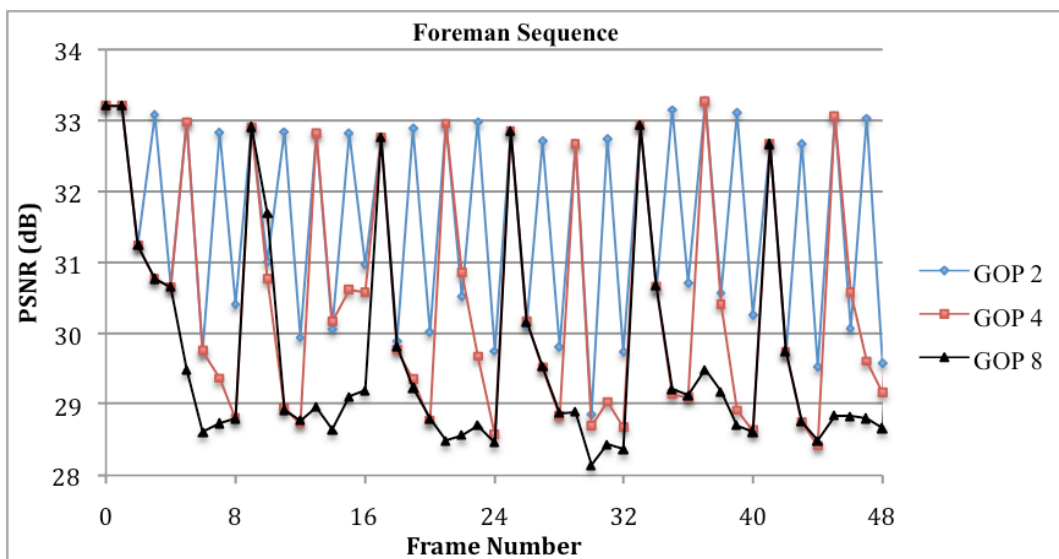


Figure 3.19 – PSNR versus Frame Number for Foreman sequence 15 Hz, using GOP size 2, 4, 8 and $Q_i = 4$.

3.5.3. WZ Frames RD Performance

For comparison purposes with the Low Delay IST DVC codec in [24], which still uses original key frames at the decoder and does not consider the rate and PSNR for the key frames, a RD performance test was performed only accounting the rate and PSNR for the WZ frames (thus excluding the rate and PSNR for the H.264/AVC Intra coded frames). These results for the Foreman sequence are presented in Figure 3.20; the test was made with the same test conditions as in [24], this means for a frame rate of 30 Hz and QCIF resolution (which is typically a more DVC friendly case than 15 Hz).

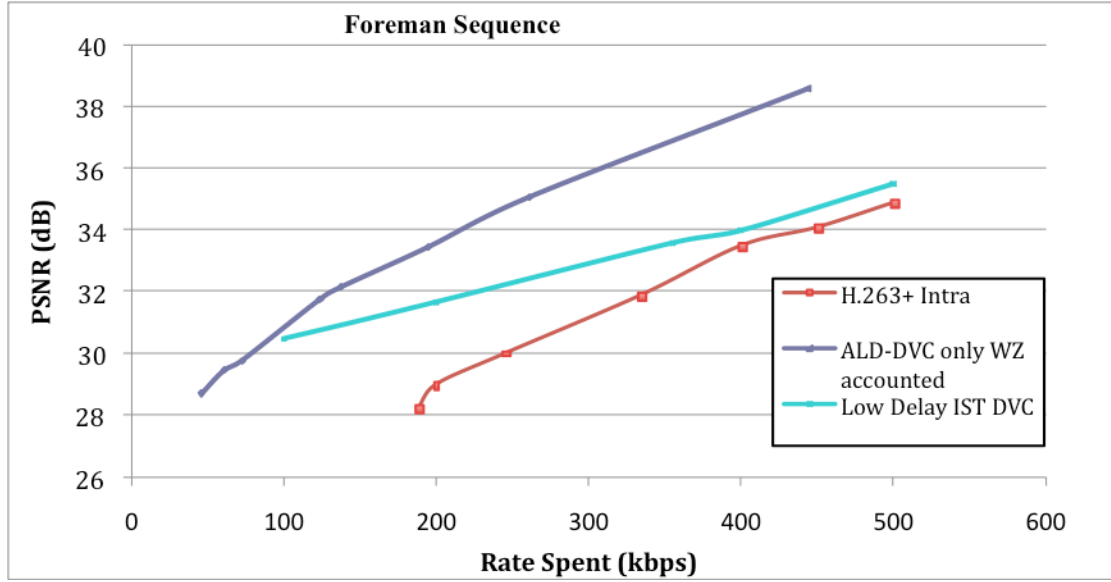


Figure 3.20 – RD performance comparison for the Foreman sequence.

According to Figure 3.20, the ALD-DVC codec presents better RD performance results than the Low Delay IST DVC codec in [24]; moreover, the RD gains increase with the bitrate, reaching almost 3 dB for the highest bitrate. These gains are motivated by:

- The fact that the ALD-DVC is a transform based codec as opposed to the pixel based Low Delay IST DVC codec; this means the ALD-DVC codec exploits the spatial redundancy and not only the temporal redundancy.
- The ALD-DVC uses 8 pixels to compute the average estimation for the uncovered areas, instead of only 3 pixels as in the Low Delay IST DVC codec.

As described above, the Low Delay IST DVC codec has an unfair advantage by using the original key frames to generate the SI frames (and not lower quality H.264/AVC coded frames), but the ALD-DVC manages to overcome this advantage by using a transform domain approach as opposed to a pixel domain approach. Any further conclusions regarding this existent gap cannot be obtained, as information pertaining other modules belonging to the Low Delay IST DVC architecture, is not provided in [24].

3.6. Final Remarks

In this chapter, it was demonstrated that the ALD-DVC codec presents good RD performance curves regarding state-of-the-art schemes such as the H.264/AVC Intra and the DISCOVER DVC codecs at least for content which is rather stable like video surveillance like content. However, there is still room for improvements, as the objective is to further close the existing RD performance gaps. In this context, a novel technique to improve the Side Information Creation process will be proposed in the next chapter, using as core architecture the ALD-DVC codec, and with the objective to further boost the RD performance of the DVC codec proposed in this chapter.

Chapter 4

Advanced Low Delay IST DVC Codec with Side Information Refinement

As explained in the previous chapters, the main objective of this Thesis is to create a practical and well performing low delay DVC codec. This means it is necessary to close the gap between the DVC codec developed and the most relevant benchmarking alternatives, both standard based and available DVC codecs. In this context, the ALD-DVC solution proposed in Chapter 3 will be improved in terms of RD performance by including a novel technique proposed by Martins in [36] which basically acts at the level of the side information creation process. The RD performance gains may imply an increase in terms of the decoder complexity, although not in a systematic way, as the encoder complexity remains the same. The basic idea of the novel decoder tool is to iteratively improve the side information along the decoding process since more information is successively available to the decoder and thus successively better side information should be generated; this approach is well known as side information refinement. While Martins proposed in [36] a side information refinement solution for the IST DVC Interpolation codec, there is no DVC solution available in the literature adopting the same refinement approach for an extrapolation-based DVC codec. In this context, the Side Information Refinement (SIR) tool will be integrated in the ALD-DVC solution proposed in Chapter 3; hence, all the remaining modules in the architecture have been already explained in Chapter 3 and, consequently, will not be discussed in this chapter unless any novelty emerges. The DVC solution to be proposed in this chapter will be labeled in the following as ALD-DVC SIR for obvious reasons. The next sections will present the novel architecture, the basic idea underlining this novel technique, the details of the several steps in the SIR algorithm and, finally, the performance evaluation and its analysis.

4.1. Codec Architecture

As mentioned above, the ALD-DVC SIR architecture presented in Figure 4.1 is very similar to the ALD-DVC architecture with the exception of the main novelty, this means the insertion of the SIR module (inside the blue box) which may increase the complexity of the decoder; this is not necessarily always the case since the improvement of the side

information obtained with the SIR tool, typically reduces the number of parity bit requests made to the encoder which contributes in the opposite direction, decreasing the decoding process complexity.

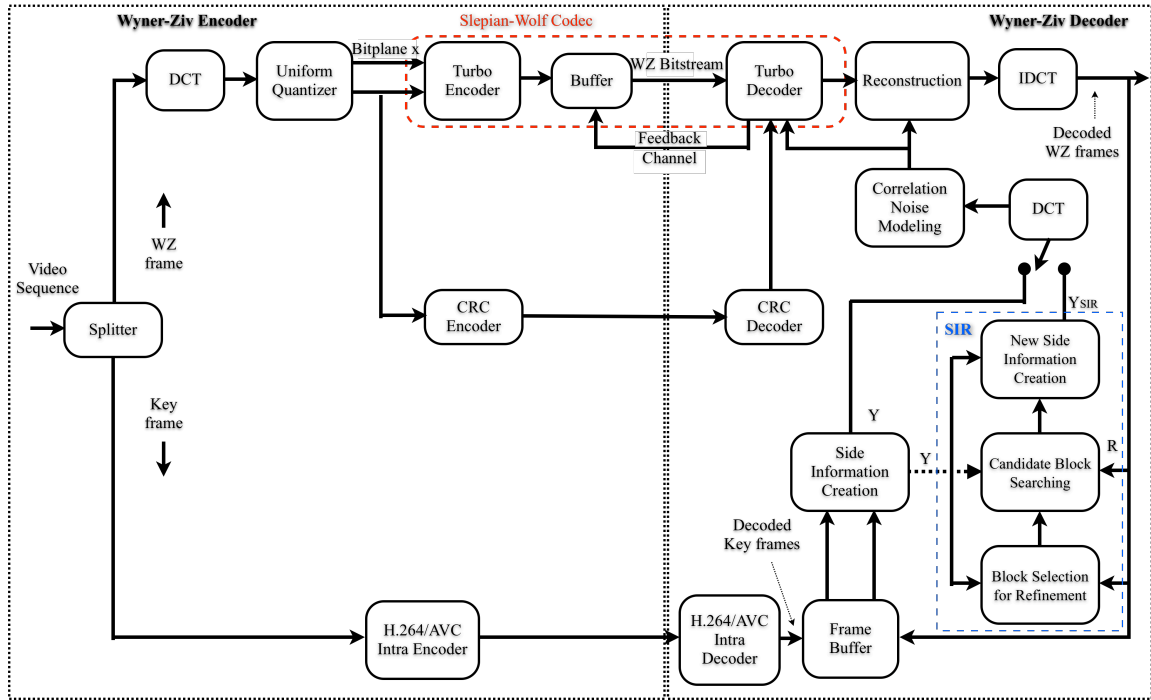


Figure 4.1 – The ALD-DVC SIR codec architecture.

The novel SIR module results from a very simple idea: by successively refining the side information along the decoding process, e.g. after each DCT band is decoded, it is possible to use successively better side information; since the quality of side information is critical for the RD performance of this type of DVC codec, this SIR approach contributes in the right direction and, thus, may help to improve the overall RD performance. Using better SI for the decoding of the next DCT bands will imply the need for a reduced WZ rate requested from the encoder to each the same final quality and, thus, better compression efficiency.

As in [36], the SIR module can be broken down into three main processing steps, notably the Block Selection for Refinement, the Candidate Block Searching and the New Side Information Creation.

- **Block Selection for Refinement** – This step decides which blocks in the side information frame need refinement after decoding each DCT band; the basic approach is, thus, to find the blocks in the side information frame which are rather different from the already decoded frame since this implies that the previously used side information was not that good and, thus, it is very likely that it may be improved.
- **Candidate Block Searching** – The second step searches for the block candidates in the initial side information, inside a given window, that may help improving the side information frame for the more efficient decoding of the next DCT bands.
- **New Side Information Creation** – Some of the candidate blocks selected in the previous step are then used to create the new side information to be used for decoding the next DCT band.

The SIR module will be presented in detail in the next section; as the other codec modules remain the same, the reader is kindly asked to refer to Chapter 3 for more detailed information.

4.2. Side Information Refinement Algorithm

In this section, the various sub-modules of the SIR algorithm will be described in detail, as they are the only novelty included the previously presented ALD-DVC architecture. The SIR tool has been implemented in this chapter according to the solution proposed by Martins [36], with no novelties introduced; however, the SIR tool acts now in an extrapolation-based architecture and not anymore interpolation-based, as Martins initially developed this tool to improve the RD performance of the IST DVC Interpolation codec. The difference is that the interpolated frames have superior quality than the ones obtained through extrapolation techniques, mainly due to motion estimation related issues.

4.2.1. Initial DCT Domain Side Information Creation

This step is not strictly part of the SIR algorithm as it corresponds to the side information creation process, already performed in the previously presented ALD-DVC architecture.

- **Initial Side Information Frame Generation** – The first iteration of the SIR algorithm only happens once for each SI frame refinement process. The first step of this entire algorithm begins with the generation of the first side information frame using the same extrapolation technique presented in the previous chapter.
- **Pixel to Transform Domain** – As soon as the initial SI frame is generated in the pixel domain, a 4×4 integer DCT transform is applied to obtain the DCT coefficients, which will serve as SI for the decoding of the first DCT band with the turbo decoder. This first DCT band is very important as it represents the DC band of the frame, band 0. All the other DCT bands needed for the reconstruction process come from the initial side information extrapolated frame since after decoding the first band only this band has been ‘corrected’. Note that the SI blocks used in the decoding process either come from the initial side information process or from the side information refinement process, as represented in Figure 4.1.

From here on, the steps presented below are successively performed after the decoding of each DCT band with the exception of the last band, which is never encoded. Nevertheless this last band is also refined as detailed below.

4.2.2. Block Selection for Refinement

As mentioned before, this sub-module has the objective to determine, after decoding each DCT band, the blocks in the SI frame able to be improved. With this target in mind, the following steps are performed:

- **Block Reconstruction** – At any given time, the block reconstruction process is the same, notably after decoding a given band $b-1$. Using the DCT bands already decoded, and copying the bands above or equal to b (not yet decoded) from the (DCT) initial side information, it is possible to reconstruct the current frame. This reconstruction begins after all the bitplanes, for a given DCT band, are decoded, reconstructing the frame and performing an inversed DCT transform to go from the transform domain to the pixel domain.

- **Error Computation** – As soon as the reconstructed current frame is available, it is necessary to assess the error between its blocks and the corresponding ones in the initial side information. It is important to note that this current frame is an already improved version of the initial side information frame, since it is already build up using some decoded and, thus, corrected DCT bands and the remaining initial side information DCT bands. Hence, by checking the error between this current frame and the SI frame blocks, using equation (4.1), it is possible to determine for each block how good is the original SI frame regarding the already decoded frame. The higher this computed error, the higher the number of errors corrected by the turbo decoder, and thus the worst the initial side information.

$$\mathcal{E}_n^b(0) = \sum_{x=0}^3 \sum_{y=0}^3 \left(Y_n(x,y) - R_n^{b-1}(x,y) \right)^2 \quad (4.1)$$

As seen in equation (4.1), $Y_n(x,y)$ and R_n^{b-1} represent the same block in the initial SI frame and the reconstructed frame, respectively, for a given block n and band $b-1$. In this context, $\mathcal{E}_n^b(0)$ is nothing more than the sum of the squared errors for the same block n after decoding band $b-1$, computed for a block size of 4×4 as described above.

- **Block Selection** – This step has the target to classify the blocks of the current frame as being good candidates to be refined or not (since it is impossible to know for sure at this stage). A block is considered a good candidate for refinement if the sum of squared errors $\mathcal{E}_n^b(0)$ computed above exceeds a certain threshold μ . The value adopted for μ , obtained through extensive experiments by Martins in [36], is 100.

4.2.3. Candidate Blocks Searching

Upon determining the blocks in the side information frame which refinement seems to be promising, it is necessary to find the SI candidate blocks that can replace and improve the SI frame quality. This second sub-module performs this analysis on the entire frame, i.e. for each 4×4 block capable of being improved:

- **Candidate Blocks Identification** – For each block with capacity to be refined, a search is performed in the initial side information frame, within a certain size window. As there is no need to search the entire frame for a candidate block, a window of $((2 \times w) + 1)$ by $((2 \times w) + 1)$ is considered, where $w = 4$. This value for w was obtained through extensive experiments, as performed to determine μ value above, simply because it represents a good trade-off between RD performance and complexity, as described in [36]. This means that there are 80 possible new SI blocks for each block selected for refinement (excluding the block chosen for refinement).
- **Matching Error Computation** – From these 80 possible candidate SI blocks, it is necessary to perform some filtering, because not all blocks are really suitable to be considered good candidate blocks. With the purpose of filtering the undesired (or not better enough) SI candidates, an error metric computed as the sum of the squared errors, between the block being refined in the reconstructed frame R_n^{b-1} and the candidate block k in the initial side information frame $Y_n^{d(k)}$ is computed, as seen in equation (4.2).

$$\mathcal{E}_n^b(k) = \sum_{x=0}^3 \sum_{y=0}^3 \left(Y_n^{d(k)}(x,y) - R_n^{b-1}(x,y) \right)^2 \quad (4.2)$$

$$d(k) = (d_x(k), d_y(k)); d_x(k), d_y(k) \in [-w, w] \quad (4.3)$$

As seen in equation (4.3), $d(k)$ is the displacement associated to the candidate block k with size 4×4 and (x, y) corresponds to the pixel position inside that same block. Note that the displacement of block k is limited by a window size of 9×9 (the 81 candidates minus the block selected for refinement itself), as $d_x(k), d_y(k) \in [-w, w]$, with $w = 4$.

- **Candidate Blocks Filtering** – Depending on error computed above, a SI candidate block is either kept or not, based on equation (4.4).

$$\epsilon_n^b(k) < \epsilon_n^b(0) \cdot (1 - P), \quad P \in [0, 1] \quad (4.4)$$

To dismiss all the SI candidate blocks that prove to be just as equal or not much better than the block ready to be refined, the sum of squared errors for a given candidate block k has to be inferior to the sum of squared errors for the initial side information block with a penalty P . The value for this penalty is considered to be $P = 0.2$, as this represents a good tradeoff between the RD performance and the additional decoder complexity.

From the 80 candidate blocks with higher error, only those fulfilling equation (4.4) will be considered eligible for the refinement process. Thus, given a certain $\epsilon_n^b(k)$ for a given block k , a weight β_n^b , as defined in equation (4.5), must be defined to assess that candidate block k correct impact for block n in the novel SI. If the sum of squared errors for a given candidate block has a low value, demonstrating that it is a good estimation, then the weight computed is increased, providing a higher confidence in that same candidate, and vice-versa.

$$\beta_n^b = \begin{cases} \frac{1}{\epsilon_n^b(k)} & \text{if } \epsilon_n^b(k) < \epsilon_n^b(0) \cdot (1 - P) \\ 0 & \text{otherwise} \end{cases} \quad (4.5)$$

4.2.4. New Side Information Creation

The final step of the refinement process is to generate the new improved SI frame, using all of the approved candidate blocks with equation (4.4). Instead of just relying on the best candidate with the lower sum of squared error value, a statistical approach is adopted using the weights β_n^b for the given candidate blocks k determined in the previous module. Thus, by using a normalized and weighted mean as in equation (4.6), a new SI for block n is obtained:

$$Y_n^{SIR}(x, y) = \frac{\sum_k (Y_n^{d(k)}(x, y) \cdot \beta_n^b(k))}{\sum_k \beta_n^b(k)} \quad (4.6)$$

Note that this whole refinement process is performed after a band $b-1$ is decoded; hence, this new refined side information frame helps decoding the next band b and obtain the new DCT coefficient values.

A) Unquantized bands processing

Depending on the quality used in the encoding process, more or less bands will be coded and sent to the decoder. The bands for which no WZ bits are sent to the decoder are called *unquantized bands*; in a normal DVC scenario, such as the

ALD-DVC codec, the missing DCT bands are simply copied from the generated initial side information frame. This is not the case with the ALD-DVC SIR codec, as this refinement process takes advantage of what is learned with the previous decoded bands. As no WZ bits are sent to help decoding the unquantized bands, there is no rate increment as the quality of the reconstructed frame is improved. Given a total of possible 16 bands, this whole process ends when the last DCT band is reached. It is important to note that even though the last band (band 16) is unquantized, the refinement process still applies to it. The next section addresses precisely the reconstruction process, both for quantized and unquantized DCT bands.

4.3. Reconstruction Algorithm

This section describes in detail the reconstruction process, essential in the refinement process above described. There are two ways to reconstruct the DCT bands/coefficients: the first involves the reconstruction of the quantized DCT bands, and the second regards the reconstruction of the unquantized DCT bands for which no WZ bits were sent from the encoder to the decoder. These unquantized DCT bands result from the use of different qualities when encoding the sequence, i.e. depending on the matrices used in the quantization process, more or less DCT bands are sent to the decoder. The main difference between the reconstruction process in the ALD-DVC SIR codec and the one used in the ALD-DVC codec is simply the number of times the reconstruction process is performed. While the ALD-DVC codec only performs the reconstruction process once for each frame, after decoding all DCT coefficients, the ALD-DVC SIR codec performs the reconstruction 16 times (corresponding to the 16 bands) for each frame.

4.3.1. Quantized DCT Bands

After successfully decoding a given DCT band in the turbo decoder, it is possible to initiate the reconstruction process. As the decoding process only gives the DCT coefficient bin q' where the original DCT coefficient should lie, it is necessary to determine a precise value for the reconstructed DCT coefficient. The reconstruction function is optimal in the sense that it minimizes the MSE of the reconstructed value for each DCT coefficient [19], if the reconstructed value for each DCT coefficient is determined as in equation (4.7):

$$x' = E[x | q', y] = \frac{\int_l^u x \cdot f_{x|y}(x | y) dx}{\int_l^u f_{x|y}(x | y) dx} \quad (4.7)$$

where x' represents the reconstructed DCT coefficient and y the corresponding DCT coefficient obtained from the SI frame. $E[.]$ represents the expected value and l and u are, respectively, the lower and upper bounds of the bin obtained from the DCT band decoding process. Furthermore, $f_{x|y}$ is the conditional probability function that models the residual statistics between the decoded DCT coefficient and the SI frame corresponding DCT coefficient. After some mathematical manipulation, the following formula is obtained:

$$x' = \begin{cases} l + b & , y < l \\ y + \frac{\left(\gamma + \frac{1}{\alpha}\right)e^{-\alpha\gamma} - \left(\delta + \frac{1}{\alpha}\right)e^{-\alpha\delta}}{2 - e^{-\alpha\gamma} - e^{-\alpha\delta}} & , y \in [l, u[\\ u - b & , y \geq u \end{cases} \quad (4.8)$$

where:

$$\begin{aligned} b &= \frac{1}{\alpha} + \frac{\Delta}{1 - e^{-\alpha\Delta}} \\ \gamma &= y - l \\ \delta &= u - y \end{aligned}$$

Here, Δ is the quantization bin size obtained from the decoding process and α the estimated alpha parameter computed in the CNM module. By observing equation (4.8), three different cases can be distinguished:

- **First Case** – This situation corresponds to the case where the side information for the DCT coefficient is below the l bound. In this case, the side information clearly failed to obtain an accurate value for that DCT coefficient. Hence, a good solution is to adopt a new value for the reconstructed DCT coefficient. Instead of simply using the bin center value, this solution computes x' using the estimated alpha parameter and the bin size, ignoring the side information DCT coefficient value, as seen in the first branch of equation (4.8).
- **Second Case** – Given a certain quantized bin obtained from the decoding process, with upper and lower bounds u and l , respectively, if the side information corresponding DCT coefficient is confined to that interval, then the value of the x' reconstructed DCT coefficient is computed using the second branch in equation (4.8). Moreover, the alpha parameter estimated in the CNM module is also used in the computation of x' , as it provides a weight, related to the confidence level of the side information DCT coefficient. An increased confidence means that the reconstructed value x' will be closer to the side information DCT coefficient value and vice-versa.
- **Third Case** – If the side information DCT coefficient value is above u , it is clear again that the side information was not accurate enough. Thus, the approach used to obtain the reconstructed x' DCT coefficient is similar to the first case, selecting a value within the interval $y \in [l, u[$ based on the estimated alpha parameter and the bin size, as seen in the third branch of equation (4.8).

4.3.2. Unquantized DCT Bands

The ALD-DVC codec described in the previous chapter, simply copies from the SI DCT coefficients the bands for which no WZ bits were sent. This implies a certain amount of problems, mainly because there are still many errors left uncorrected in these specific bands. The ALD-DVC SIR codec applies exactly the same process but adds on the top of it the novel refinement process, which boosts the RD performance as no WZ rate is spent and only gains are achieved by refining the SI frame.

4.4. Performance Evaluation

The test conditions for the performance evaluation of the ALD-DVC SIR codec proposed in this chapter are exactly the same as described in Chapter 3 and, thus, will not be repeated here.

4.4.1. RD Performance for GOP Size 2

The RD performance results for the ALD-DVC SIR codec using GOP size 2, regarding all the tested sequences are presented in Table 4.1.

Table 4.1 – RD performance results for ALD-DVC SIR codec.

Coastguard		Foreman		Hall Monitor		Soccer	
Rate (kbps)	PSNR (dB)	Rate (kbps)	PSNR (dB)	Rate (kbps)	PSNR (dB)	Rate (kbps)	PSNR (dB)
84.04	27.8	70.59	28.6	86.28	31.01	52.46	27.73
103.39	28.46	84.24	29.35	96.38	31.6	65.46	28.45
108.42	28.57	93.87	29.93	100.07	31.67	75.61	29.27
167.1	30.37	145.69	32.51	137.16	34.06	128.03	31.86
185.54	30.89	152.54	32.59	140.92	34.07	134.36	31.91
249.3	32.14	197.87	33.92	176.58	35.7	180.35	33.09
297.91	33.16	257.91	35.99	204.99	37.14	246.99	35.04
496.67	36.37	405.69	39.3	307.78	40.51	451.33	39.02

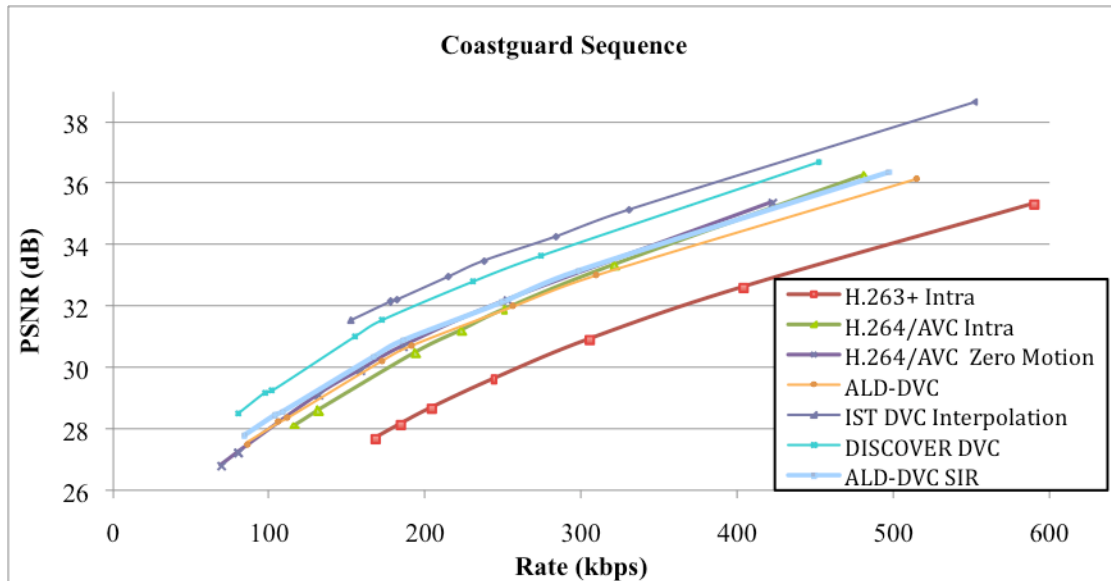


Figure 4.2 – RD performance comparison for the Coastguard sequence using GOP size 2.

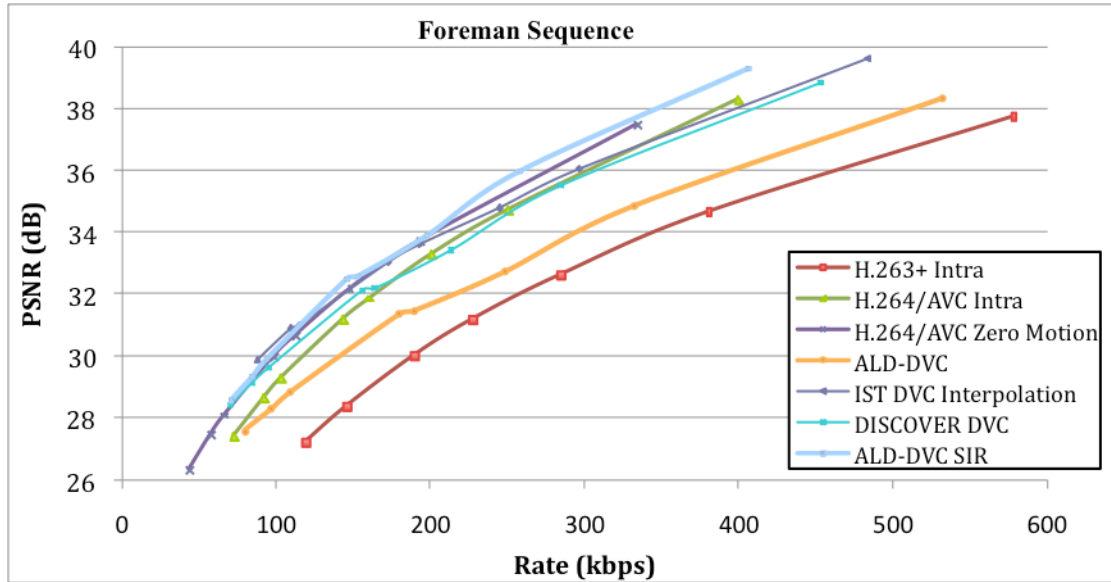


Figure 4.3 – RD performance comparison for the Foreman sequence using GOP size 2.

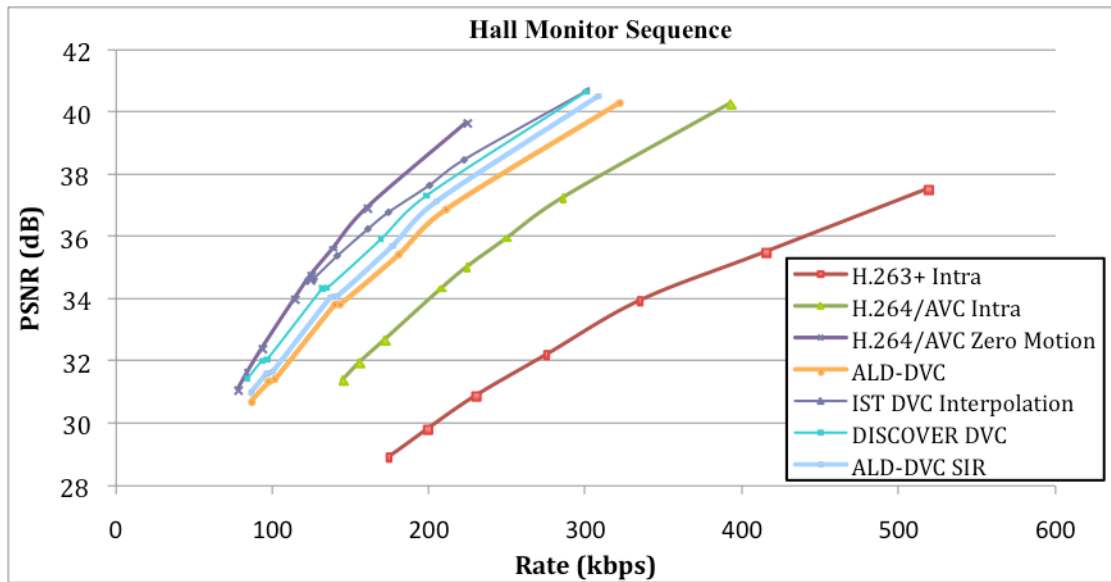


Figure 4.4 – RD performance comparison for the Hall Monitor sequence using GOP size 2.

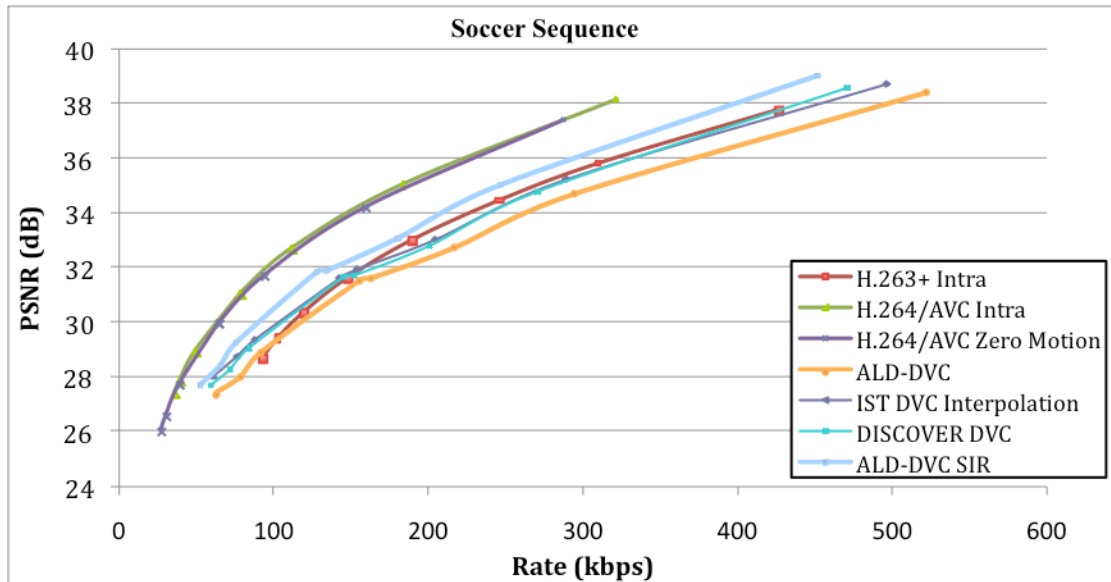


Figure 4.5 – RD performance comparison for the Soccer sequence using GOP size 2.

- ALD-DVC SIR versus ALD-DVC** – Although the comparison with other more common benchmarking solutions presented below is important, this analysis is the most relevant one as it shows the gains obtained by evolving from the ALD-DVC codec to the ALD-DVC SIR codec; this means the RD performance benefits from the use of a SI refinement approach. As mentioned before, only rate gains are expected since the same quantization matrices are used; thus, the only question regards the amount of rate reduction obtained by using the SIR module, which corresponds to a RD performance gain for a certain rate. Bearing that in mind, it can be concluded that the Coastguard and Hall Monitor sequences present the lowest RD performance gains; this is expectable as they are the most stable sequences, presenting rather low complexity motion and, thus, the initial side information is already rather good and little refinement may be made. For the Coastguard sequence, the SIR module does not bring any relevant gains, as both curves seem to be almost overlapping, slightly diverging for the last RD points where the gain never exceeds 0.3 dB in the best case; for this case, the trade-off between RD performance and decoding complexity very likely does not justify the adoption of the SIR tool, at least for the lower rates. The Hall Monitor sequence also shows reduced gains only up to 0.4 dB but still better than for the Coastguard sequence, proving that the ALD-DVC SIR codec already performs better than the ALD-DVC codec. The ALD-DVC SIR codec definitely shows better RD performance results for the more complex motion sequences, notably Foreman and Soccer. In the Soccer sequence, gains up to 1 dB are registered for the higher RD points and, thus, higher rates and qualities. The best results regard the Foreman sequence as the ALD-DVC SIR codec shows improvements up to almost 3dB. This was expected since when the motion is more complex, the side information extrapolation process is less reliable and, thus, the initial side information may be largely improved with the novel information obtained with the successive decoding of the DCT bands.
- ALD-DVC SIR versus IST DVC Interpolation** – For the Coastguard and Hall Monitor sequences, as the ALD-DVC SIR codec behaves almost in the same way as the ALD-DVC codec, the analyses performed in the previous chapter still apply. The IST DVC Interpolation codec performs better for these sequences with its RD performance curve always appearing above the ALD-DVC SIR codec RD performance curve. This behavior is inverted for the other two

tested sequences. For the Foreman sequence, the ALD-DVC SIR codec performs rather well increasing the gap between these two codecs as the rate grows; for the higher RD points corresponding to the higher qualities, the gains go up to 1.5 dB. Regarding the Soccer sequence, the ALD-DVC SIR codec RD performance curve remains always above the IST DVC Interpolation RD performance curve, showing gains of almost 1 dB. This fact implies that the gains obtained with the usage of the SIR tool in an extrapolation-based codec are higher than the gains obtained with the usage of an interpolation-based SI creation solution, which has delay as an additional cost.

- **ALD-DVC SIR versus DISCOVER DVC Codec** – As described in the previous chapter, the IST DVC Interpolation codec is an evolution of the DISCOVER DVC codec, always presenting better results in terms of RD performance for most sequences, namely the Coastguard, Foreman and Hall Monitor sequences (and very similar for Soccer). Thus, whatever the sequence analyzed, the gap between the ALD-DVC SIR codec and the DISCOVER DVC codec can only be smaller when compared to the gap between the ALD-DVC SIR and IST DVC Interpolation codecs (for Soccer the gains are similar).
- **ALD-DVC SIR versus standard Intra codecs** – When comparing the ALD-DVC SIR codec against the H.263+ Intra standard one thing is clear: the ALD-DVC SIR codec developed by the author of this Thesis surpasses this standard based coding scheme for every sequence and condition. For the Coastguard sequence, the gains obtained are constant as both RD distortion curves present a constant gap of around 2 dB. The largest gains occur for the Foreman and the Hall Monitor sequences with gains up to 4 dB and 6.5 dB, respectively. Regarding the Soccer sequence, the gains registered are still relevant but rather low when considering the other sequences gains; these gains go up to 0.8 dB for the highest rates, mainly because this is a very high complexity motion sequence. When considering the state-of-the-art H.264/AVC Intra codec, the scenario is not as good but still shows significant improvements with the ALD-DVC SIR codec. In the most difficult sequence, the Soccer sequence, the ALD-DVC SIR codec performs below the H.264/AVC Intra codec with losses up to 1.5 dB. The RD performance gap closes when considering the Coastguard sequence as both curves, the ALD-DVC SIR and the H.264/AVC Intra codecs, overlap almost perfectly. When considering the last two sequences, the ALD-DVC SIR codec RD performance curves overcome the Intra coding standard scheme. For the Foreman sequence, the best case shows a gain of 1 dB, favoring the codec developed in this thesis. The gap widens even more for the Hall Monitor sequence, with the ALD-DVC SIR codec achieving a gain of up to almost 2.5 dB. In summary, the SIR tool allows reducing the losses or increasing the gains of the DVC codec regarding the H.264/AVC Intra codec depending on the previous situation with the ALD-DVC codec.
- **ALD-DVC SIR versus standard Inter Zero Motion codecs** – As described in the previous chapter, the Inter Zero Motion codecs exploit part of the temporal redundancy, increasing the RD performance notably for sequences with low complexity motion, such as the Hall Monitor sequence, since there are large static areas. Thus, for that particular sequence, the ALD-DVC SIR codec still registers losses up to 2 dB, as the H.264/AVC Zero Motion codec already greatly explores the redundancy between frames. For the Soccer sequence, the ALD-DVC SIR codec still performs under the predictive scheme but the losses are smaller, this means 1.2 dB for the worst case scenario. For the Coastguard sequence, a perfect overlapping between these two codecs is visible without any relevant gains or losses for each. Finally, the Foreman sequence presents the best results with gains almost up to 0.5 dB.

As expected, the ALD-DVC SIR codec only presents gains regarding the previous ALD-DVC codec demonstrating that it is a better DVC solution when compared with the adopted benchmarking solutions. Depending on the content of the sequence, the ALD-DVC SIR codec proposed by the author of this Thesis performs better or worse; however, it seems clear that for stable and medium complex motion content, the proposed DVC solution already performs in a very promising way.

4.4.2. RD Performance for Longer GOP Sizes

Similarly to the previous chapter, it is also important to analyze the ALD-DVC SIR codec RD performance using longer GOP sizes, notably 4 and 8, to check the consistency of the RD performance. It is important to notice that the larger the GOP size, the lower is the overall encoder complexity since the WZ frames encoding process is less complex than the key frames encoding process. These results are presented below in Figures 4.6, 4.7, 4.8 and 4.9. For a better benchmarking, the charts include also the ALD-DVC codec results for the same GOP sizes to show the gains obtained with the introduction of the SIR approach.

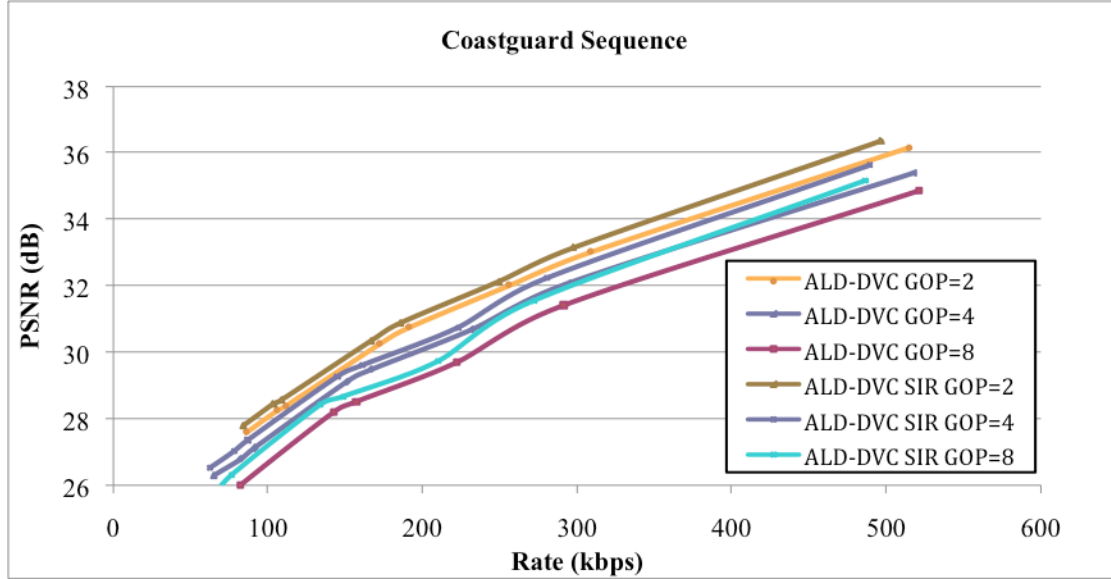


Figure 4.6 – RD performance comparison for the Coastguard sequence using GOP size 2, 4 and 8.

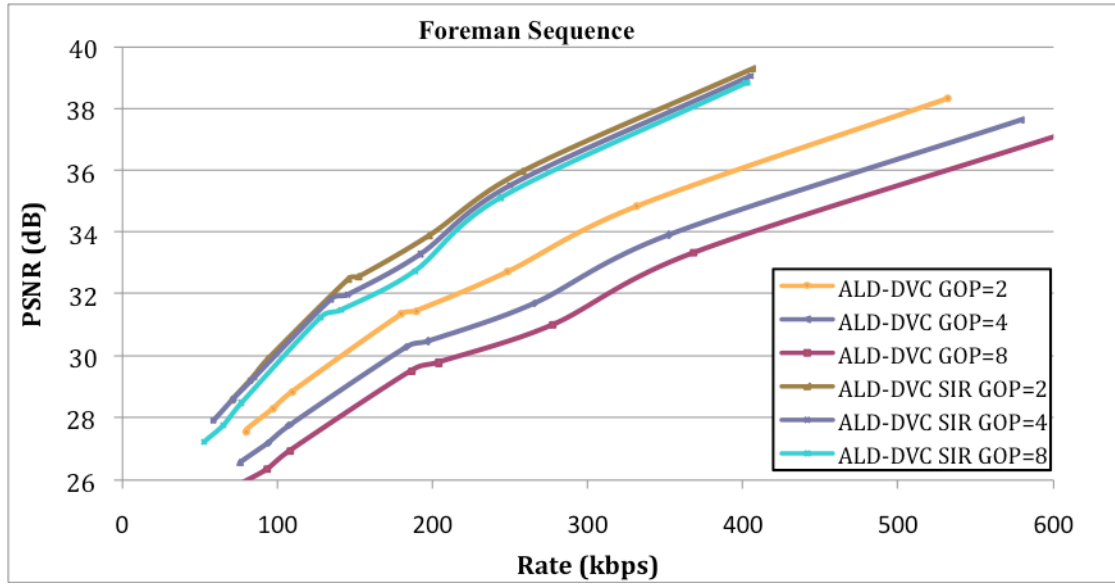


Figure 4.7 – RD performance comparison for the Foreman sequence using GOP size 2, 4 and 8.

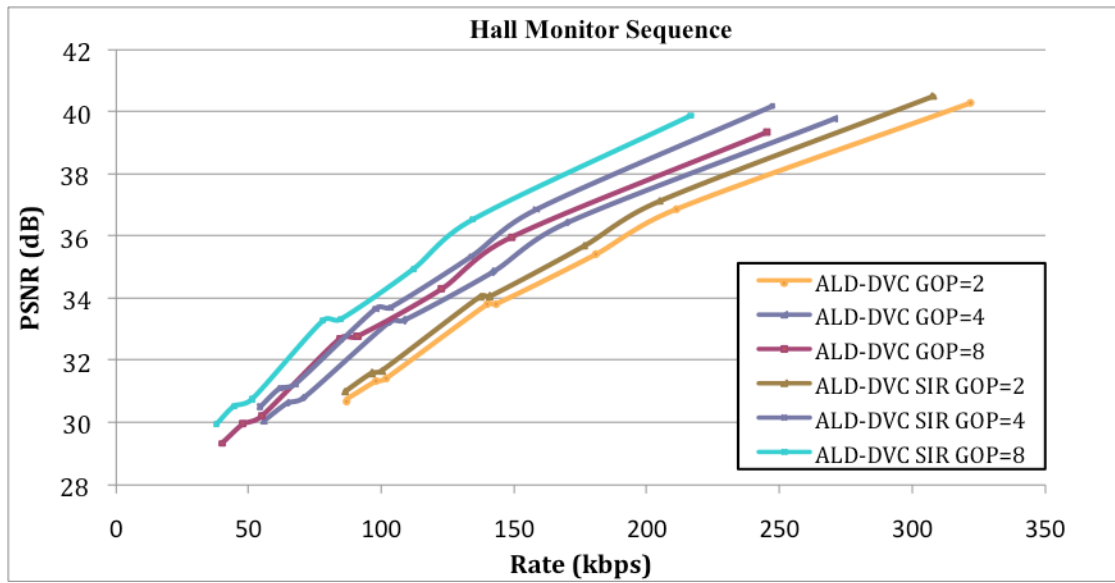


Figure 4.8 – RD performance comparison for the Hall Monitor sequence using GOP size 2, 4 and 8.

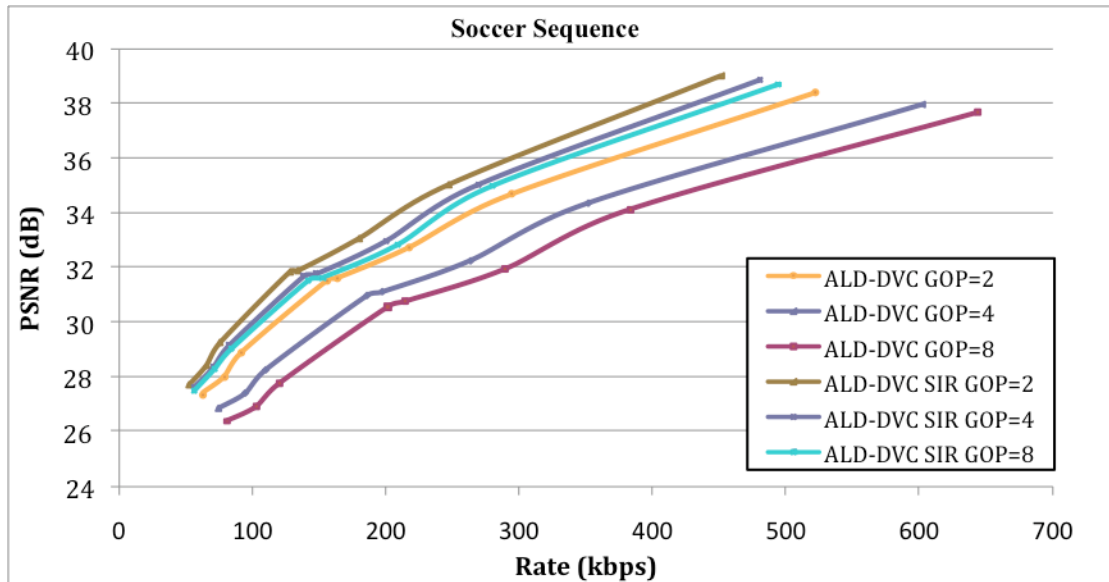


Figure 4.9 – RD performance comparison for the Soccer sequence using GOP size 2, 4 and 8.

- ALD-DVC SIR codec performance for different GOP sizes** – As expected, as the GOP size increases there is a drop in the RD performance for almost every sequence, except for the Hall Monitor sequence which is the most stable. The lack of Intra coded key frames with increased quality over the WZ frames takes a toll in the performance of the ALD-DVC codecs both with and without SIR. The reason for this behavior was already discussed in the previous chapter when studying a similar situation. With lower quality decoded frames used in the extrapolation process, there is a growing drop in the reliability of the motion estimation process and the side information quality as the GOP size increases. As there is motion in the Coastguard, Foreman and Soccer sequence, this expected behavior is observed in Figures 4.6, 4.7, 4.8 and 4.9. For the Hall Monitor sequence, which is characterized by very low motion, the RD performance for the ALD-DVC SIR codec grows with the GOP size. The refinement process allied to the good motion estimation performed by the Side Information Creation module is very efficient for this particular sequence. The gains, obtained for GOP size 8, are high enough to overcome even the predictive scheme benchmarking H.264/AVC Zero Motion codec, as seen in Figure 4.10. Some benchmarking solutions were removed from the charts as they were already presented in Figure 4.4 and they are irrelevant here. For the Foreman sequence, there is almost no drop in the RD performance curves as the GOP size increases from 2 to 8 which seems to indicate that the SIR tools compensates the lower quality of the initial side information which is not the case for the ALD-DVC codec.
- ALD-DVC SIR versus ALD-DVC using same GOP sizes** - The RD performance comparison between the ALD-DVC and the ALD-DVC SIR codecs using the same GOP sizes seems to indicate that the ALD-DVC SIR codec gains typically increase with the GOP size. The largest gains are obtained for the Foreman sequence where the gains for GOP size 4 go up to 4 dB, and for GOP size 8 go up to 5 dB.

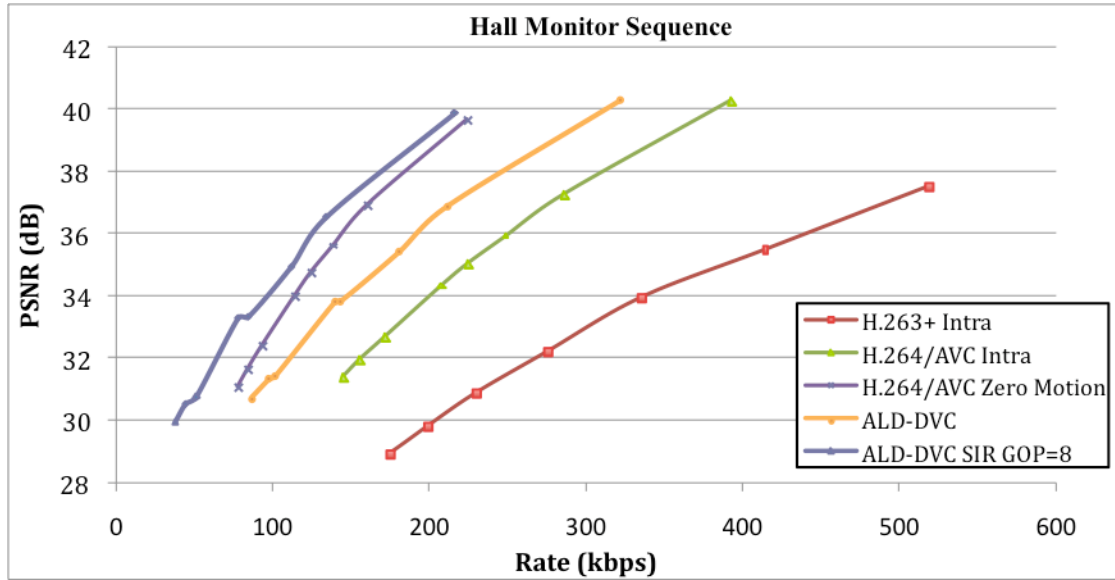


Figure 4.10 – RD performance comparison for Hall Monitor sequence using GOP size 8.

4.5. Final Remarks

As demonstrated in this chapter, the ALD-DVC codec proposed in Chapter 3 performs even better with the inclusion of the SIR module, providing more promising RD performance results. For specific types of content, such as surveillance content, the RD performance is competitive regarding the most relevant standard-based benchmarks even for longer GOP sizes where the low complexity benefits are larger. This is even more important considering that this type of content corresponds precisely to the type of applications where the low encoding complexity and low delay requirements are more critical.

Chapter 5

Conclusions and Future Work

The audio-video coding world has been continuously changing in the last two decades, although always essentially based in the predictive coding paradigm. However, the traditional coding paradigm used for years does not seem to fit very well some of the emerging applications needs. For example, the usual paradigm of complex encoders versus, simple decoders is very much adequate for digital TV video services, as the encoder needs to send a decodable stream to multiple (thousands) of cheaper and less complex decoders. However, important emerging applications, such as wireless video surveillance systems, do not seem to be well addressed with a traditional system where (many) high cost cameras with powerful encoders transmit to a simple and low cost decoder, since this be considered a very poor investment.

The need for a different coding paradigm was considered, notably one that would shift complexity from the encoder to the decoder. In 2002, based on two theorems developed in the 70's, namely the Slepian-Wolf theorem [3] and the Wyner-Ziv theorem [4], a new type of practical video coding solutions has emerged, creating a whole new field well known as distributed video coding (DVC). In this context, Chapter 1 had the purpose to contain all the useful information to help understand this new coding paradigm, notably the theoretical foundations for the DVC video coding branch. To provide context and background to the work developed in this Thesis, Chapter 2 had the objective to propose a classification taxonomy to organize the current range of DVC solutions available. With this purpose, four classification dimensions were defined, appropriate to classify any DVC solution. Note that other classification dimensions could have been considered but those adopted in Chapter 2 help to better understand the objectives and the work developed by the author of this Thesis. In the context of this classification framework, several types of solutions were presented, notably state-of-the-art DVC schemes, such as the DISCOVER DVC solution [12], and others due to their relevance for the development of the codec in this Thesis.

Given the relevant background, it was possible to understand better the work described in Chapter 3, as well as the results obtained for the proposed ALD-DVC codec. Moreover, the RD performance results demonstrated that the ALD-DVC codec is not only low delay driven, since it uses side information extrapolation based techniques, but it is also rather efficient as it performs better or at least equally to some standard based schemes, such as H.263+ Intra and even the very efficient

H.264/AVC Intra codec. It was also shown that the ALD-DVC codec performs very well when compared to the state-of-the-art DVC schemes, such as the DISCOVER DVC codec, further closing the gap between extrapolation and interpolation based DVC codecs.

To further improve the low delay DVC performance, a new module, called Side Information Refinement module (SIR), was integrated in the ALD-DVC codec, as explained in Chapter 4 based on the idea of continuously refining the initial side information along the decoding process. This new module boosts the RD performance, as important gains are reached, tightening the existent gap between standard and DVC schemes. At given times, the ALD-DVC SIR codec was so efficient, notably for GOP size 8, that even the H.264/AVC Zero Motion RD distortion curve was surpassed for the low motion complexity Hall Monitor sequence.

As demonstrated above, the ALD-DVC SIR codec proposed in Chapter 4 is an evolution of the ALD-DVC codec, detailed in Chapter 3. As newer and better techniques are developed every day, it is safe to say that the ALD-DVC SIR is still open for improvements, thus able to reach RD performance results even better than the ones presented in Chapter 4. These improvements may be obtained by enhancing many of the ALD-DVC SIR modules, supporting any new ideas/algorithms, substituting or simply changing the already available algorithms. Naturally, the efficiency and low delay requirements have always to be considered but it is quite possible to create an even better version of the ALD-DVC SIR codec by developing research in the following areas:

- **Improved Extrapolation Techniques** – As interpolation techniques have an advantage in terms of the side information quality due to the usage of future frames, the extrapolation based motion estimation techniques have to be more innovative when only using past frames. This effect may be reached by using more past frames such as the solution presented in [22], which is able to capture the true motion and create a better motion vector field. Another idea for future work regarding the ALD-DVC motion estimation, i.e. building the motion vector field, is to adopt a certain threshold value for the WMAD computation, and use only the various candidate blocks with a WMAD value below that threshold to compute a new motion vector. All the block candidate positions meeting the requirements would be considered in the motion vector creation, with higher or lower weights, depending on the WMAD value for that specific candidate block. Another viable improvement may be to capture the true motion using the 3-D Recursive Search algorithm presented in [38] with the possible use of the Content Adaptive Resolution tool described in [34]. All these possible improvements would contribute to raise the compression efficiency of the codec developed in this Thesis.
- **Improved Key Frames** – It is largely accepted that standard codecs include very refined tools since they have been developed and improved for a significant number of years. The state-of-the-art H.264/AVC Intra codec, also used in the ALD-DVC codec, is the result of many years of research. The H.264/AVC Intra codec is a very good solution when coding the key frames, mainly because it provides great quality spending almost half the rate of previous standards, at the expense of increased processing complexity. Improvements in this area can also be brought to the DVC codecs, if there is still the need to encode key frames in an efficient way, thus increasing the ALD-DVC codec efficiency.
- **Improved Correlation Noise Modeling** – The correlation noise model developed for the ALD-DVC codec is not the definitive way to characterize the statistics between the WZ and SI frames differences. Some alternatives were

implemented regarding this module and the presented one selected, since it reached the best results. Naturally, further developments in this area would provide even greater efficiency not only to the ALD-DVC codec but also to all the DVC codecs using low delay online correlation noise estimation at the decoder.

- **Improved Channel Codes** – The channel codes are a very important step in the DVC architecture as they help to correct the errors in the generated side information frame. The channel codes considered in the ALD-DVC codec were the Turbo Codes, well known in the DVC literature. The Low-Density Parity-Check codes (LDPC) were also available to use in the ALD-DVC codec, but its usage was dismissed due to the high computational complexity. There are also important developments in this area, notably the so-called Raptor Codes [39], which may provide even greater gains, increasing the efficiency of the developed codec.

Even though it was not mentioned above, the increase in efficiency motivated by the implementation of all these new ideas would reduce the stress on the feedback channel, a major constraint in adopted DVC architecture, as fewer errors would need to be corrected and thus less rate would be asked from the encoder. Naturally, based on the conclusions presented above, it is plausible to accept the ALD-DVC SIR as a good solution within the DVC field, considering the fact that it fulfils, in a rather promising way, the objectives proposed at the beginning of this Thesis: compression efficiency and low delay.

References

- [1] A. Aaron, R. Zhang and B. Girod, “Wyner-Ziv Coding of Motion Video”, Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, November 2002.
- [2] A. Wyner, “Recent Results in the Shannon Theory”, IEEE Trans. on Information Theory, vol. 20, n° 1, pp. 2 - 10, January 1974.
- [3] J. Slepian and J. Wolf, “Noiseless Coding of Correlated Information Sources”, IEEE Trans. on Information Theory, vol. 19, no 4, pp. 471 - 480, July 1973.
- [4] A. Wyner and J. Ziv, “The Rate-Distortion Function for Source Coding with Side Information at the Decoder”, IEEE Trans. on Information Theory, vol. 22, no 1, pp. 1 - 10, January 1976.
- [5] A. Aaron, S. Rane, E. Setton and B. Girod, “Transform-Domain Wyner-Ziv Codec for Video”, Visual Communications and Image Processing Conference, San Jose, California, USA, January 2004.
- [6] B. Girod, A. Aaron, S. Rane and D. Rebollo Monedero, “Distributed Video Coding”, Proceedings of the IEEE, vol. 93, n° 1, pp. 71 - 83, January 2005.
- [7] R. Puri and K. Ramchandran, “PRISM: A New Robust Video Coding Architecture Based on Distributed Compression Principles”, 40th Allerton Conference on Communication, Control and Computing, Allerton, IL, USA, October 2002.
- [8] R. Puri, A. Majumdar and K. Ramchandran, “PRISM: A Video Coding Paradigm with Motion Estimation at the Decoder”, IEEE Trans. on Image Processing, vol. 16, n° 10, pp. 2436 - 2448, October 2007.
- [9] A. Kubota, A. Smolic, M. Magnor, M. Tanimoto, T. Chen, C. T. Zhang, “Multiview Imaging and 3DTV”, IEEE Signal Processing Magazine, vol. 24, n° 6, pp. 10-21, November 2007.
- [10] C. Brites, F. Pereira, “Encoder rate Control for Transform Domain Wyner-Ziv Video Coding”, International Conference on Image Processing, San Antonio, TX, USA, September 2007.

- [11] J. D. Areia, J. Ascenso, C. Brites, F. Pereira, “Low Complexity Hybrid Rate Control for Lower Complexity Wyner-Ziv Video Decoding”, European Signal Processing Conference, Lausanne, Switzerland, August 2008.
- [12] *DISCOVER Project Page*, <http://www.img.lx.it.pt/~discover/home.html>
- [13] X. Artigas, F. Tarres, L. Torres, “Comparison of Different Side Information Generation Methods for Multiview Distributed Video Coding”, International Conference on Signal Processing and Multimedia Applications, Barcelona, Spain, July 2007.
- [14] C. Brites, J. Ascenso and F. Pereira, “Studying Temporal Correlation Noise Modeling for Pixel based Wyner-Ziv Video Coding”, IEEE International Conference on Image Processing, Atlanta, GA, USA, October 2006.
- [15] C. Brites and F. Pereira, “Correlation Noise Modeling for Efficient Pixel and Transform Domain Wyner-Ziv Video Coding”, IEEE Trans. on Circuits and Systems for Video Technology, vol. 18, n° 9, pp. 1177-1190, September 2008.
- [16] J. Ascenso, C. Brites and F. Pereira, “Improving Frame Interpolation with Spatial Motion Smoothing for Pixel Domain Distributed Video Coding”, EURASIP Conf. on Speech and Image Processing, Multimedia Communications and Services, Smolenice, Slovak Republic, June 2005.
- [17] A. Trapanese, M. Tagliasacchi, S. Tubaro, J. Ascenso, C. Brites and F. Pereira, “Intra Mode Decision based on Spatio-Temporal Cues in Pixel Domain Wyner-Ziv Video Coding”, International Conference on Acoustics, Speech, and Signal Processing, Toulouse, France, May 2006.
- [18] D. Varodayan, A. Aaron and B. Girod, “Rate-Adaptive Codes for Distributed Source Coding”, EURASIP Signal Processing Journal, Special Issue on Distributed Source Coding, vol. 86, n° 11, pp. 3123 - 3130, November 2006.
- [19] D. Kubasov, J. Nayak and C. Guillemot, “Optimal Reconstruction in Wyner-Ziv Video Coding with Multiple Side Information”, IEEE Multimedia Signal Processing Workshop, Chania, Crete, Greece, October 2007.
- [20] F. Pereira, C. Brites, J. Ascenso, “Distributed video coding: basics, codecs and performance”, chapter in the book “Distributed source coding: theory, algorithms and applications”, edited by Michael Gastpar and Pier Luigi Dragotti, Academic Press, 2009.
- [21] S. Borchert, R. P. Westerlaken, R. Klein Gunnewiek, R.L. Lagendijk, “Improving Motion Compensated Extrapolation for Distributed Video Coding”, Thirteenth Annual Conference of the Advanced School for Computing and Imaging, Heijen, The Netherlands, June 2007.
- [22] S. Borchert, R. P. Westerlaken, R. Klein Gunnewiek, R.L. Lagendijk, “On Extrapolating Side Information in Distributed Video Coding”, 26th Picture Coding System, Lisbon, Portugal, November 2007.
- [23] A. Aaron, S. Rane, E. Setton and B. Girod, “Transform-Domain Wyner-Ziv Codec for Video”, Visual Communications and Image Processing Conference, San Jose, CA, USA, January 2004.

- [24] L. Natário, C. Brites, J. Ascenso, F. Pereira, "Extrapolating Side Information for Low-Delay Pixel-Domain Distributed Video Coding", Int. Workshop on Very Low Bitrate Video Coding, Sardinia, Italy, September 2005.
- [25] W.A.R.J. Weerakkody, W.A.C. Fernando, J.L. Martínez, P. Cuenca, F. Quiles, "An Iterative Refinement Technique for Side Information Generation in DVC", IEEE International Conference on Multimedia and Expo, Beijing, China, July 2007.
- [26] A.B.B. Adikari, W.A.C. Fernando, W.A.R.J. Weerakkody, "Side Information Improvement in DVC with Two Side Information Streams and 3D Motion Refinement", Canadian Conference on Electrical and Computer Engineering, Vancouver, Canada, April 2007.
- [27] A.B.B. Adikari, W.A.C. Fernando, W.A.R.J. Weerakkody, "Multiple Side Information Streams for Distributed Video Coding," IET Electronics Letters, vol. 42, Issue 25, pp. 1447-1449, March 2006.
- [28] A.B.B. Adikari, W.A.C. Fernando, W.A.R.J. Weerakkody, H.K.Arachchi, "Sequential Motion Estimation using Luminance and Chrominance Information for Distributed Video Coding of Wyner-Ziv Frames", IEE Electronics Letters, vol. 42, Issue 7, pp. 398- 399, March 2006.
- [29] C. Tomasi, R. Manduchi, "Bilateral Filtering for Gray and Color Images", Sixth International Conference on Computer Vision, Bombay, India, January 1998.
- [30] J. Ascenso, C. Brites, F. Pereira, "Content Adaptive Wyner-Ziv Video Coding Driven by Motion Activity", IEEE International Conference on Image Processing, Atlanta, GA, USA, October 2006.
- [31] M. Tagliasacchi, S. Tubaro, A. Sarti, "On the Modeling of Motion in Wyner-Ziv Video Coding", IEEE International Conference on Image Processing, Atlanta, GA, USA, October 2006.
- [32] Z. Li, L. Liu, and E. J. Delp, "Rate Distortion Analysis of Motion Side Estimation in Wyner-Ziv Video Coding", IEEE Trans. on Image Processing, vol. 16, n° 1, pp. 98-113, January 2007.
- [33] M. Tagliasacchi, A. Trapanese, S. Tubaro, J. Ascenso, C. Brites, F. Pereira, "Exploiting Spatial Redundancy In Pixel Domain Wyner-Ziv Video Coding", IEEE International Conference on Image Processing, Atlanta, GA, USA, October 2006.
- [34] R. Braspenning, G. de Haan, "Efficient Motion Estimation with Content-Adaptive Resolution", Proceedings of International Symposium on Consumer Electronics, pp. 29-34, September 2002.
- [35] C. Brites, J. Ascenso, J. Pedro, F. Pereira, "Evaluating a feedback channel based transform domain Wyner-Ziv video codec", Signal Processing: Image Communication, vol. 23, no. 4, pp. 269-297, April 2008.
- [36] R. Martins, J. Ascenso, C. Brites, F. Pereira, "Refining Side Information for Improved Transform Domain Wyner-Ziv Video Coding", IEEE Transactions on Circuits and Systems for Video Technology.

- [37] T. Wiegand, G. J. Sullivan, G. Bjntegaard and A. Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Trans. Circuits Systems for Video Technology, vol.13, n°7 pp.560-576, July 2003.
- [38] Haan, G. deet al, "True-Motion Estimation Using 3-D Recursive-Search Block Matching", IEEE Trans. on Circuits and Systems for Video Technology, vol. 3, n° 5, October 1993.
- [39] H. Pishro-Nik and F. Fekri, "On Raptor Codes," in Proceedings of the IEEE International Conference on Communications (ICC '06), vol. 3, pp. 1137–1141, June 2006.