TÉCNICO LISBOA

Statistical Reconstruction for Predictive Video Coding

Vitor Hugo Nunes Gomes

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Prof. Fernando Manuel Bernardo Pereira

Prof. João Miguel Duarte Ascenso

Examination Committee

Chairperson: Prof. Fernando Duarte Nunes

Supervisor: Prof. João Miguel Duarte Ascenso

Member of the Comitee: Prof. José Manuel Peixoto Nascimento

May 2014

Acknowledgments

First, I would like to thank Professor Fernando Pereira, my supervisor, for allowing me to work on this Thesis and his constant guidance and supervising. His suggestions and advices influenced in great deal the outcome of this work. It was a privilege working under his supervision and the amazing work ethic and methodology that he displayed has set an example for me to strive for in future work.

I would like to express my gratitude to Professor João Ascenso, my co-supervisor, for the availability to address my questions during the work process and always provide meaningful and important advices. His conceptual and technical knowledge was vital to the work developed in this Thesis.

I am also thankful to Professor Catarina Brites for stepping in at a crucial time in the development of the Thesis. Her help in the latter stages, even when it interfered with her schedule, was instrumental for the developing of the final work presented.

I would like to thank both my parents and my sister for supporting me in life and in my academic journey from day one. Without their unconditional love and support, and the examples they have set for me, my life would certainly not achieve this stage.

I would also like to thank my beloved girlfriend Catarina Rodrigues, for all the support both at a personal level, and in my academic journey. Her constant emotional support keeps me grounded and on track. Thank you for always being there for me.

Finally to all my friends at Instituto Superior Técnico, André Martins, Daniel Santos, João Guerreiro, João Halm, João Oliveira, João Paiva, João Piedade, João Tiago Fonseca, Miguel Alexandre, Nuno Fontes, Pedro Santos, Ricardo Martins, Ricardo Póvoa, Roberto Tomé, Vasco Escaleira and Vasco Nascimento a big thank you for the companionship and good times spent throughout my college experience and all your help.

Thank you all.

Abstract

As for the previous video coding standards, the H.264/AVC (Advanced Video Coding) standard adopts a predictive coding paradigm combining temporal prediction with a spatial transform, quantization and entropy coding to achieve good rate-distortion (RD) performance. The complexity associated to this process lies mostly at the encoder side, keeping the decoder as simple as possible. On the other hand, the alternative Distributed Video Coding (DVC) approach proposes to exploit the video redundancy mostly at the decoder side, keeping the encoder as simple as possible. One of the most characteristic DVC tools is the statistical reconstruction of the Discrete Cosine Transform (DCT) coefficients, which plays a similar role as the inverse scalar quantization in predictive codecs. The main objective of this Thesis is to study the use of statistical reconstruction as a substitute to inverse quantization in the context of the H.264/AVC standard, thus creating a video coding architecture with a mix of predictive and distributed coding tools.

After reviewing the relevant literature on the H.264/AVC standard with emphasis on the quantization process, and on DVC solutions with emphasis on the statistical reconstruction, a statistical reconstruction tool has been developed to replace the usual inverse scalar quantization adopted in the H.264/AVC standard. This solution adopts a Laplacian correlation model for the DCT residuals and estimates the model parameter to best fit the DCT residuals. The statistical reconstruction tool is integrated on both the encoder and decoder sides and uses the Laplacian model to improve the process of reconstructing the quantized DCT coefficients.

Experimental results obtained using the Bjontegaard (BD) metric show BD-Rate savings up to 4,71% and a BD-PSNR increase up to 0,33 dB, when comparing the proposed solution with the reference software H.264/AVC codec. When comparing the proposed solution with the H.264/AVC+ARO (Adaptive Rounding Offsets) video codec using a offset control based quantization, BD-Rate savings and BD-PSNR gains are observed for the lower bitrates, while at higher bitrates the proposed solution has a slight performance disadvantage.

Keywords: H.264/AVC; Predictive Video Coding; Distributed Video Coding; Scalar Dequantization; Statistical Reconstruction.

Sumário

Tal como nas normas de codificação de vídeo anteriores, a norma H.264/AVC (*Advanced Video Coding*) adopta um paradigma preditivo de codificação de vídeo combinando predição temporal com transformada espacial, quantização e codificação entrópica para atingir um bom desempenho débito-distorção. A complexidade associada a este processo situa-se maioritariamente do lado do codificador, mantendo-se o descodificador o mais simples possível. Por outro lado, a abordagem alternativa designada como Distributed Video Coding (DVC), propõe-se explorar a redundância do vídeo essencialmente do lado do descodificador, mantendo o codificador tão simples quanto possível. Uma das ferramentas características do DVC é a reconstrução estatística dos coeficientes DCT que tem um papel semelhante à quantização inversa nos codecs preditivos. O principal objectivo desta Tese é estudar o uso da reconstrução estatística como substituto da quantização inversa no contexto da norma H.264/AVC, criando assim uma arquitectura de codificação de vídeo com uma mistura de ferramentas dos tipos preditivos e distribuídos.

Após a revisão da literatura relevante sobre a norma H.264/AVC com ênfase no processo de quantização, e sobre as soluções DVC com ênfase na reconstrução estatística, foi desenvolvida uma ferramenta de reconstrução estatística para substituir a habitual quantização inversa adoptada na norma H.264/AVC. Esta solução adopta um modelo de correlação Laplaciano para o resíduo dos coeficientes DCT e estima o parâmetro do modelo que melhor se adequa a esse resíduo. Esta ferramenta de reconstrução estatística é integrada tanto do lado do codificador como do lado do descodificador e usa o modelo Laplaciano para melhorar o processo de reconstrução dos coeficientes DCT quantizados.

Os resultados experimentais obtidos usando a métrica de Bjontegaard (BD) evidenciaram uma poupança em BD-Rate até 4,71% e um aumento em BD-PSNR até 0,33 dB, quando a solução proposta foi comparada com a norma H.264/AVC. Ao comparar a solução proposta com a norma H.264/AVC com *Adaptive Rounding Offsets* (ARO) activos, foram observadas poupanças em BD-Rate e ganhos em BD-PSNR para os débitos binários mais baixos, enquanto que para os débitos binários mais elevados a solução proposta tem uma ligeira desvantagem.

Palavras-chave: H.264/AVC; Codificação Distribuída de Vídeo; Quantização Inversa; Reconstrução Estatística.

Table of Contents

Chapter 1 - Context and Objectives 1

1.1. Context and Motivation 1

1.2. Objectives 2

1.3. Report Organization 3

Chapter 2 - Reviewing Background Technology 5

2.1. Reviewing Predictive Video Coding 5

2.1.1. Basic Concepts 5

2.1.2. The State-of-the-Art H.264/AVC Standard 7

2.1.2.1. Network Abstraction Layer 8

2.1.2.2. Video Coding Architecture 9

2.1.2.3. Main Novel Coding Tools 10

2.1.2.4. Performance Assessment 13

2.2. Reviewing Distributed Video Coding 14

2.2.1. Basic Concepts and Early Wyner-Ziv Video Coding Solutions 14

2.2.2. The DISCOVER Wyner-Ziv Video Codec 16

2.2.2.1. Architecture and Walkthrough 16

2.2.2.2. Performance Assessment 17

2.3. Reviewing Relevant Background on Quantization 19

2.3.1. Basic Issues on Scalar Quantization 19

2.3.2. Transform and Quantization in H.264/AVC 21

2.3.2.1. H.264/AVC Integer Transform Design 22

2.3.2.2. H.264/AVC Quantization Process 23

2.3.3. Adaptive Quantization Algorithms 25

2.3.3.1. Adaptive Quantization using an Equal Expected-Value Rule 25

2.3.3.2. Adaptive Quantization based on Rounding Offsets 27

2.4. Reviewing Relevant Background on Correlation Noise Modeling and Optimal Reconstruction 31

2.4.1. Correlation Noise Modeling for Efficient Transform Domain Wyner-Ziv Video Coding 31
2.4.2. Optimal Reconstruction in WZ Video Coding with Multiple Side Information ... 35
Chapter 3 - Predictive Video Coding with Statistical Reconstruction: Video Codec Architecture ... 39
Chapter 4 - Predictive Video Coding with Statistical Reconstruction: Novel Coding Tools ... 43
4.1. Optimal Transform, Scaling and Quantization ... 43
4.2. Residual Statistical Modeling ... 44
4.2.1. DCT Coefficients Statistical Analysis ... 44
4.2.2. Statistical Model Parameter Computation ... 48
4.3. Statistical Reconstruction ... 49
4.3.1. DCT Coefficients Reconstruction Bin Bounds Computation ... 49
4.3.2. DCT Coefficients Reconstruction ... 51
Chapter 5 - Predictive Video Coding with Statistical Reconstruction: Performance Assessment ... 53
5.1. Test Conditions ... 53
5.1.1. Video Sequences ... 53
5.1.2. Coding Conditions ... 55
5.1.3. Performance Evaluation Metrics ... 56
5.2. RD Performance Evaluation ... 57
Chapter 6 - Concluding and Future Work ... 67
6.1. Summary ... 67
6.2. Achievements ... 68
6.3. Future Work ... 68
References ... 71
Annex A - Studying the Performance of the ARO Algorithm ... 73

Index of Figures

Figure 1 - Scope of video coding standardization [1]. 7
Figure 2 - Structure of H.264/AVC standard [1]. 8
Figure 3 - Basic H.264/AVC encoder architecture [1]. 9
Figure 4 - Multi-frame motion compensation example for a P-slice [6]. 11
Figure 5 - Example of deblocking filter performance: without (left) and with (right) deblocking filter [7] 12
Figure 6 - Graphical representation of the H.264/AVC profiles [8] 13
Figure 7 – RD performance and bitrate savings plot for entertainment-quality applications [7] 13
Figure 8 - DISCOVER Wyner–Ziv video codec architecture [9]. 16
Figure 9 - RD performance comparison for GOP size 2 [10]. 18
Figure 10 - RD performance comparison for GOP sizes 2, 4 and 8 [10]. 19
Figure 11 – Example of a quantizer structure [12]. 21
Figure 12 - Performance comparison between encoders with and without adaptive rounding method for the sequence Mobile CIF (left) and Soccer 4CIF (right) [12]. 27
Figure 13 - Block diagram for the ARO algorithm [14]. 29
Figure 14 - Relative control errors for the “erin” (left) and “royal” (right) sequences: (a)(b) I in ‘III’; (c)(d) P in ‘IPP’; (e)(f) B in ‘IBP’ [14]. 30
Figure 15 - Decoder Structure with Multiple Side Information [17]. 37
Figure 16 - RD performance with optimal MMSE reconstruction [17]. 38
Figure 17 - High-level encoder architecture of the proposed video coding solution. 41
Figure 18 – High-level decoder architecture of the proposed video coding solution. 42
Figure 19 - Residual histogram and Laplacian fitting for the I frames AC1 band of the sequence City, 1280x720, 60 Hz, QP = 10. 45
Figure 20 - Residual histogram and Laplacian fitting for the P frames AC5 band of the sequence City, 1280x720, 60 Hz, QP = 11. 45
Figure 21 - Residual histogram and Laplacian fitting for the B frames AC9 band of the sequence City, 1280x720, 60 Hz, QP = 12. 46
Figure 22 - Residual histogram and Laplacian fitting for the I frames AC1 band of the sequence Night, 1280x720, 60 Hz, QP = 10. 46
Figure 23 - Residual histogram and Laplacian fitting for the P frames AC5 band of the sequence Night, 1280x720, 60 Hz, QP = 11. 47
Figure 24 - Residual histogram and Laplacian fitting for the B frames AC9 band of the sequence Night, 1280x720, 60 Hz, QP = 12. 47

Figure 25 – First frame of the selected video sequences: top) Night (left) and City (right); bottom) Big Ships (left) and Shuttle Start (right) 54
Figure 26 – RD performance comparison for the Night sequence: lower rates 57
Figure 27 – RD performance comparison for the Night sequence: higher rates 58
Figure 28 - RD performance comparison for the City sequence: lower rates 58
Figure 29 - RD performance comparison for the City sequence: higher rates 59
Figure 30 - RD performance comparison for the Shuttle Start sequence: lower rates 59
Figure 31 - RD performance comparison for the Shuttle Start sequence: higher rates 60
Figure 32 - RD performance comparison for the Big Ships sequence: lower rates 60
Figure 33 - RD performance comparison for the Big Ships sequence: higher rates 61
Figure 34 – RD performance comparison for the Coastguard CIF sequence 74
Figure 35 – RD performance comparison for the Foreman CIF sequence 74
Figure 36 – RD performance comparison for the Mobile CIF sequence 75
Figure 37 – RD performance comparison for the Soccer CIF sequence 75
Figure 38 – RD performance comparison for the Night 720p sequence 76
Figure 39 – RD performance comparison for the Big Ships 720p sequence 76

Index of Tables

Table 1 - Δ and PSNR comparison for HD contents [14]. 30
Table 2 - DCT Band-Level and Coefficient-Level RD performance for Flower Garden, Foreman, Coastguard and Hall Monitor QCIF Sequences [16]. 35
Table 3 – Test video sequences characteristics 55
Table 4 – Quantization parameters used to define each RD point for the various video sequences (RD points 1-4) 55
Table 5 - Quantization parameters used to define each RD point for the various video sequences (RD points 5-9) 56
Table 6 – Bjontegaard metric results for H.264/AVC + Statistical Reconstruction vs H.264/AVC: higher rates 63
Table 7 – Bjontegaard metric results for H.264/AVC + Statistical Reconstruction vs H.264/AVC: lower rates 63
Table 8 – Bjontegaard metric results for H.264/AVC + Statistical Reconstruction vs H.264/AVC + ARO: higher rates 64
Table 9 - Bjontegaard metric results for H.264/AVC + Statistical Reconstruction vs H.264/AVC + ARO: lower rates 64

List of Acronyms

ARO	Adaptive Rounding Offset
AVC	Advanced Video Coding
BD	Bjontegaard
CABAC	Context-Adaptive Binary Arithmetic Coding
CAVLC	Context-Adaptive Variable Length Coding
CIF	Common Intermediate Format
CNM	Correlation Noise Model
DCT	Discrete Cosine Transform
DVC	Distributed Video Coding
FI	Frame Interpolation
FMO	Flexible Macroblock Ordering
FQ	Forward Quantizer
GOP	Group of Pictures
HEVC	High Efficiency Video Coding
HD	High Definition
HVS	Human Visual System
IDCT	Inverse Discrete Cosine Transform
IDR	Instantaneous Decoding Refresh
IQ	Inverse Quantization
ITU-T	International Telecommunications Union – Telecommunication Standardization Sector
JVT	Joint Video Team

LDPC	Low Density Parity Check
MCTI	Motion Compensated Temporal Interpolation
MPEG	Moving Picture Experts Group
MSE	Mean Squared Error
NAL	Network Abstraction Layer
NURQ	Nearly-Uniform Reconstruction Quantizer
PAFF	Picture-Adaptive Frame/Field
PCM	Pulse-Code Modulation
PRISM	Power-Efficient Robust High-Compression Syndrome-Based Multimedia
PSNR	Peak-Signal-to-Noise-Ratio
QCIF	Quarter Common Intermediate Format
RD	Rate Distortion
SI	Side Information
SD	Standard Definition
URQ	Uniform Reconstruction Quantizer
VCEG	Video Coding Experts Group
VCL	Video Coding Layer
WZ	Wyner-Ziv

Chapter 1

Context and Objectives

This chapter intends to present the overall scope and objectives of this Thesis along with its motivation and context. Finally, the Thesis structure is presented.

1.1. Context and Motivation

Nowadays, digital video has a regular and well established presence in our lives. Digital television, personal computers and handheld devices, such as smart phones, are now fully integrated in our society, and are used extensively to access, record and play digital videos. The growth in digital visual content usage has been accompanied with the development of powerful compression tools that enable the reduction of the bitrate necessary to represent these contents by exploiting data correlation and the limitations of the human visual system (HVS) to remove redundant and irrelevant data, respectively. These coding tools have been included in several video coding standards defined by the International Telecommunications Union – Telecommunication Standardization Sector (ITU-T) and the Moving Picture Experts Group (MPEG) over the last two decades. Currently, the H.264/AVC standard [1], developed by the Joint Video Team (JVT) formed by the ITU-T Video Coding Experts Group (VCEG) and ISO/IEC MPEG standardization groups, is the most deployed video coding solution in the market but a new video coding standard has already been defined, the so-called High Efficiency Video Coding (HEVC) standard [2]. The HEVC standard offers again 50% bitrate reductions for the same perceptual quality and should start conquering the markets soon.

In the available video coding standards, all adopting a predictive coding paradigm combining temporal prediction with a spatial transform and quantization, substantial RD gains have been achieved by increasing the encoder complexity while maintaining the decoder complexity the lowest possible. This coding paradigm is well adapted to some very important video coding applications like television broadcasting and video streaming which follow a one-to-many topological model with a single complex encoder providing coded content to multiple less complex decoders. In this coding paradigm, the encoder complexity is typically five to ten times larger than the decoder complexity [3]. The encoder complexity is mainly associated with the motion estimation process, which is responsible for the creation of efficient predictions and thus for the high RD performance achieved.

However, emerging applications like wireless video surveillance, multimedia sensor networks and mobile camera phones, amongst others, are challenging the usual predictive coding paradigm, notably in terms of complexity allocation. For example, in wireless video surveillance systems, low complexity encoders are desired, since there is typically a high number of encoders which should be simple and only one or a few decoders which can be more complex than the encoders. Distributed video coding (DVC) is a new video coding paradigm, emerged around 2002, that fully or partly exploits the video redundancy at the decoder and not anymore at the encoder as in predictive video coding; with this complexity balance, DVC codecs are well suited for some emerging application scenarios as those mentioned above. According to the Slepian-Wolf theorem [4], it is possible to achieve the same bitrate as required for the typical joint encoding and decoding (with a vanishing error probability) as used in predictive coding by independently encoding but jointly decoding two statistically dependent signals. The Wyner-Ziv theorem extends the Slepian-Wolf theorem to the lossy case (in practice, more relevant), becoming the key theoretical basis for Wyner-Ziv (WZ) video coding where a source is lossy coded based on the availability of some correlated source at the decoder from which is derived the so-called *side information* (SI) [4]. The side information is an estimation of the original frame (source) to code made at the decoder based on already decoded information. Since the estimation process naturally includes some errors (this means the side information is different from the original frame), the encoder has the task to 'correct' the estimation errors in a similar way that errors in an error-prone channel are corrected, this means using a channel code, thus obtaining a decoded frame that has better quality than the estimated side information frame.

While many well know tools used in the predictive video coding paradigm have been integrated in distributed video coding solutions, it became clear at some stage that also some tools typically used in distributed video coding may be used in predictive video codecs. These tools may be integrated with the expectation of improving the predictive video coding RD performance, eventually also slightly increasing the encoder and decoder complexity. Exploiting the synergy between the two video coding paradigms, predictive and distributed is the main goal of this Thesis.

1.2. Objectives

In the context defined above, the main objective of this Thesis is to enhance the overall RD performance of the state-of-the art H.264/AVC video codec by integrating a technique typically used by distributed video decoders, the statistical reconstruction which has a similar purpose to the inverse quantization used in predictive codecs such as H.264/AVC. To reach this target, the following tasks had to be performed:

- Review the relevant literature on both predictive and distributed video coding, notably contributions related to inverse quantization in predictive video coding and statistical reconstruction in distributed video coding.
- Design a predictive video coding architecture where a statistical reconstruction method improves the inverse quantization (IQ) process as adopted in distributed video decoders.
- Define a correlation model for the H.264/AVC residual DCT coefficients. In the predictive video coding context, the decoded frame is obtained by adding the Intra/Inter predicted frame (which is obtained with the coding modes/motion vectors sent by the encoder) to the residual frame (which is obtained by entropy decoding followed by IQ and IDCT of the

residual). Since in distributed video coding the side information is an estimation of the original frame to be coded, it corresponds to the Intra/Inter predicted frame in predictive video coding. Thus, the correlation between the frame to be coded and the side information must be statistically modeled to adopt a DVC based statistical reconstruction method to replace the H.264/AVC decoder inverse quantization. Although, this process may increase the H.264/AVC encoder and decoder complexity, it may also bring advantages, notably RD performance benefits.

- Estimate the parameters for the defined correlation noise model with fine granularity to have a dynamically enough adaptation to the evolving data statistics.
- Design and implement an appropriate decoder statistical reconstruction method able to improve the RD performance of the H.264/AVC video codec.
- Evaluate the RD performance improvements obtained with the proposed video coding method against the relevant benchmarks.

1.3. Report Organization

This report is organized in six chapters, including this first chapter that is used to introduce the work to be developed and precisely define its objectives.

Chapter 2 contains a review of the state-of-the art on the relevant aspects of video coding technologies, notably predictive and distributed video coding. With this review, the reader is introduced to some basic principles and tools necessary for a better understanding of the work to be developed. Especial focus will be given to the quantization in predictive codecs and statistical reconstruction in distributed codecs.

In Chapter 3, the proposed video codec architecture is presented together with a brief walkthrough of the main innovations proposed in this Thesis, namely the new modules inserted in the H.264/AVC video codec.

In Chapter 4, the video coding tools proposed in this Thesis are presented in detail in order to provide deep insight on the effective work done.

In Chapter 5, the evaluation of the proposed coding solution in terms of RD performance is presented and the obtained results are analyzed.

Finally, in Chapter 6, the main conclusions about the work and possible future developments in this area are presented.

Chapter 2

Reviewing Background Technology

This chapter has the main objective to review the background technologies relevant for this Thesis. In this context, the first section is dedicated to a predictive video coding overview, notably the basic concepts and its most popular representative, this means the H.264/AVC standard; this standard will be first briefly described, notably its architecture and main tools and, finally, its RD performance will be presented. The second section will be dedicated to briefly review distributed video coding providing some background on its basics and a concise description of the DISCOVER video codec, the most important benchmark for this video coding paradigm. In the third section, some review on quantization and inverse quantization will be provided and, finally, in the fourth section a brief review on correlation noise modeling and optimal reconstruction will be made. These topics together build the basis for the video codec to be designed, implemented and evaluated in this Thesis.

2.1. Reviewing Predictive Video Coding

This section intends to provide an overview on predictive video coding and its main tools and characteristics since this is the coding paradigm adopted by all video coding standards developed in the past 20 years. A brief description of the state-of-the-art H.264/AVC video codec will be provided, notably its architecture, main tools and performance.

2.1.1. Basic Concepts

Nowadays, there is an exponential growth in the number of services using digital video, and an increasing popularity of high definition (HD) contents. Video coding for telecommunication applications has evolved through the development of the ITU-T H.261, H.262 (MPEG-2 Video), H.263 (and later enhancements, known as H.263+ and H.263++) and MPEG-4 video coding standards. Throughout this evolution, continued efforts have been made to maximize the compression efficiency while dealing with a large diversity of networks and their characteristics, while always also keeping in mind the loss/error robustness, delay, random access and complexity requirements. The practical source coding/compression problem may be described as follows: given a maximum allowed delay and a maximum allowed complexity, an optimal

tradeoff has to be achieved between the bitrate and the distortion/quality for the range of network environments and conditions envisioned by the relevant applications [5].

Video coding typically uses a representation system with three components: Y which is called *luminance* (also luma) and represents the brightness, and C_b and C_r which are the two *chrominance* components and represent the extent to which the color deviates from the luminance towards blue and red, respectively [5]. As it is a known fact that the human visual system is more sensitive to variations in the luminance component, the video codecs often take advantage of this fact by using what is called a colour subsampled format where the chrominance spatial resolution is lower than the luminance spatial resolution; for example, in a 4.2:0 subsampling, the chrominances have half the resolution in both directions, vertical and horizontal, which means that the chrominance component arrays have only one fourth of the samples of the corresponding luminance array.

Pulse-code modulation (PCM) is the simplest form of digital source coding where each sample is independently represented with the same number of bits, typically 8 bits per sample for image and video data. As the simplest digital representation, it is also the least efficient in terms of bitrate and so compression becomes necessary to reduce the involved bitrates.

The basic idea in predictive video coding is to exploit the redundancy and irrelevancy present in the PCM video data to reduce the coding rate while reaching a certain target quality. While exploiting the irrelevancy implies eliminating from the signal representation information that cannot be recovered making the codec a lossy codec (this means the decoded and original videos are mathematically different), exploiting the redundancies implies eliminating repeated/redundant information, in time, space and statistics and, thus, does not bring any degradation to the decoded signal regarding the original PCM signal representation. The basic tools used in predictive video coding solutions such as the previously mentioned standards are:

- **Temporal Prediction**: Process where a set of prediction values is created for each video frame that is used to predict the values of the original samples so that only the differences from the prediction values need to be coded; this tool exploits the temporal redundancy in the video as it is possible to create a low prediction error or residue to code each image by exploiting its similarities with previously coded images. In order to improve the temporal prediction, motion estimation and compensation tools are commonly used. These tools have the target of improving temporal predictions for each image area, by detecting, estimating and compensating the motion in the various video regions. The objective of motion estimation is to relate the position of a given sample, area or object in an image with its position in a previously coded image in the past or in the future. The difference between the positions in the current and previous/future frame(s) constitutes the motion vector which has to be coded and sent to the decoder.
- **Spatial Transformation**: Using a spatial transformation can avoid to repeatedly represent similar sample values as it is possible to capture the essence of the signal by using frequency analysis. The spatial transform´s objective is to obtain an alternative representation of the signal while offering more effective coding. This tool exploits the statistical correlation of the input samples, so that the most relevant information associated to the set of input samples is typically concentrated into a small set of values, the so-called *transform coefficients*.

- **Quantization**: To exploit the visual irrelevance in the original signal, the transform coefficients to transmit for each block are quantized, thus introducing some mathematical error. This process is the main responsible for the quality losses in DCT based codecs but also bitrate reductions. Each quantization step is selected taking into account the minimum perceptual difference for the coefficient in question as this determines the associated (if any) negative quality impact. For example, the HVS is less sensitive to high spatial frequencies and very low/high luminances. Taking this into account, a coarser coding of these coefficients can be made through the use of quantization, thus improving the overall compression as no bits are wasted to code non-perceptually perceivable information.
- **Entropy Coding**: A process where the relative probabilities of the various possible values of each source symbol are exploited through a suitable compressed. This means the existent statistical redundancy in the source is used to represent the video data, notably the quantized transform coefficients and motion vectors.

Naturally, video data can be compressed by simply coding each picture independently, leading to the so-called *Intra coding approach.* However, it is possible to achieve much improved compression performance by taking advantage of the large temporal redundancy in video content, leading to the so-called *Inter coding approach*. By exploiting the spatial and temporal redundancy, only the video data that cannot be appropriately predicted from previously coded data is sent. The ability to exploit the temporal redundancy to improve the compression efficiency is what distinguishes video compression from pure Intra compression, such as used for example in the JPEG standard. Due to its compression performance compared with other video coding technologies, predictive video coding has been adopted in all video coding standards developed by MPEG and ITU-T since the beginning of the nineties, notably the H.261, H.263, MPEG-1, MPEG-2 and MPEG-4 standards.

2.1.2. The State-of-the-Art H.264/AVC Standard

The H.264/AVC standard represents currently the state-of-the-art on video coding providing about 50% increased compression efficiency regarding the previously available standards. To achieve this improved RD performance, H.264/AVC includes some key enhancement tools when compared to prior video coding methods that will be briefly presented in the following. As shown in Figure 1, the scope of video coding standardization is only the bitstream and the decoder as the encoder does not need to be specified to provide interoperability. This limitation of the standardization scope enables freedom to optimize the encoder implementations in a manner appropriate to each specific application; however, no guarantees on the decoded quality are provided [1].

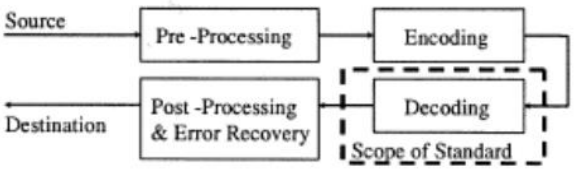


Figure 1 - Scope of video coding standardization [1].

To provide flexibility and customizability, the H.264/AVC design covers a Video Coding Layer (VCL), which is designed to efficiently represent the video content, and a Network Abstraction Layer (NAL), which formats the VCL representation of the video and provides header

information in an suitable way for transmission by a variety of network technologies and storage media as represented in Figure 2.

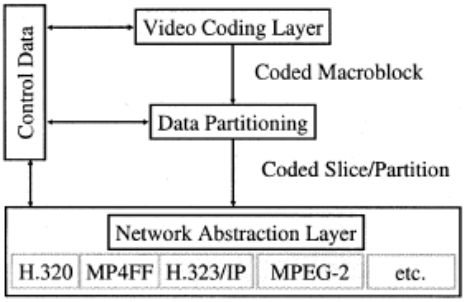


Figure 2 - Structure of H.264/AVC standard [1].

2.1.2.1. Network Abstraction Layer

The NAL is designed to provide network friendliness, thus enabling a simple and effective customization of the VCL data for a broad variety of systems. H.264/AVC is not customized to fit the needs of any particular application or network but the design of the NAL anticipates a variety of such mappings. The video coded data is organized into NAL units, each corresponding to a packet containing an integer number of bytes. The first byte of each NAL unit is a header byte indicating the type of data in the NAL unit while the remaining bytes contain payload data. Some emulation prevention bytes with a specific value, called a *start code prefix,* may be inserted in the data to prevent a particular pattern of data from accidentally being generated. NAL units are classified into VCL and non-VCL NAL units. The VCL NAL units contain the data that represent the values of the samples in the video pictures, while the non-VCL NAL units contain any associated metadata information such as *parameter sets* and *supplemental enhancement information*.

Parameter sets contain important header information that is valid for a large number of VCL NAL units. There are two types of parameter sets: the *sequence parameter sets* and the *picture parameter sets*. The first contains parameters of the coded video sequence, while the second contains parameters essential for the decoding of individual pictures. Each VCL NAL unit has an identifier referring to the content of the relevant picture parameter set and each picture parameter set has an identifier referring to the content of the relevant sequence parameter set. This enables referring to a large amount of information using only the identifier. Sequence and parameter sets can be sent ahead of the VCL NAL units that they apply to provide robustness against data loss.

NAL units are grouped together to form an *access unit*. The decoding of each access unit results in one decoded picture. A coded video sequence consists on a series of access units that are sequential in the NAL unit stream and use only one sequence parameter set. Each coded video sequence can be decoded independently of any other coded video sequence, given the necessary parameter set information, conveyed “in-band” or “out-of-band”. At the beginning of a coded video sequence, there is an instantaneous decoding refresh (IDR) access unit. An IDR access unit contains an intra-picture which is a coded picture that can be decoded without decoding any previous pictures in the NAL unit stream; the presence of an IDR access unit indicates that no subsequent picture in the stream will reference pictures prior to the intra picture in order to be decoded.

2.1.2.2. Video Coding Architecture

The VCL design follows the usual block-based predictive video coding approach, where each coded picture is represented in block-shaped units of associated luminance and chrominance samples called *macroblocks* with a size of 16×16 luminance samples. The luminance and chrominance samples of a macroblock are either spatially or temporally predicted, and the resulting prediction residual is encoded using transform coding. Each color component of the prediction residual signal is subdivided into smaller 4×4 blocks. Each block is transformed using an integer transform, and the transform coefficients are quantized and encoded using entropy coding methods. Figure 3 presents the basic H.264/AVC encoder architecture which main modules will be described in the following.

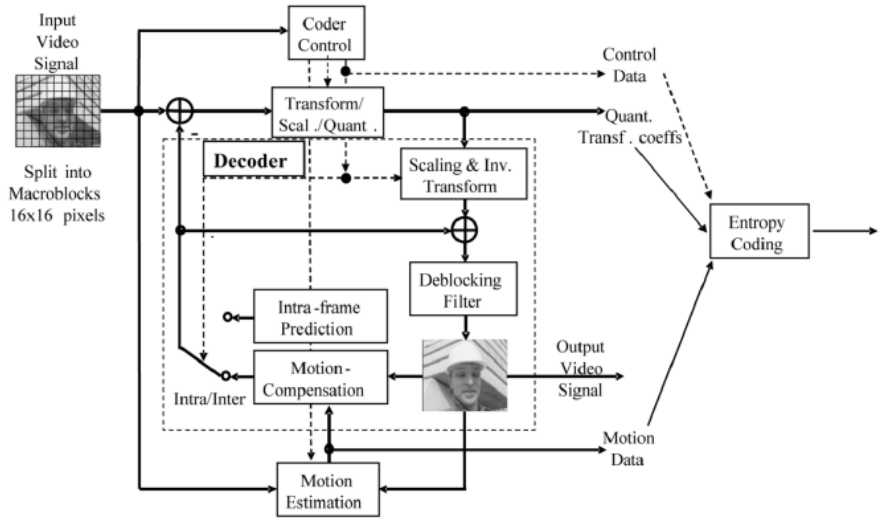


Figure 3 - Basic H.264/AVC encoder architecture [1].

In terms of data representation and structuring, the H.264/AVC standard has the following main features:

- **Y C_b C_r Color Space and Sampling -** Initially, H.264/AVC exploited the HVS characteristics by using a 4:2:0 sampling with 8 bits per sample, meaning that the chrominances are coded with ¼ of the spatial resolution. As the standard evolved, other subsampling formats have been accepted in later profiles.
- **Picture Partitioning into Macroblocks -** A picture is partitioned into fixed-sized macroblocks, each covering a rectangular picture area of 16×16 samples of the luma component and 8×8 samples of each of the two chroma components for 4:2:0 subsampling. Macroblocks are the basic frame building blocks for which the standard specifies the bitstream syntax and semantics and the decoding process.
- **Slices and Slice Groups -** Slices are sets of macroblocks which are processed in raster scan order. They are self-contained and their syntax elements can be parsed from the bitstream without any previous knowledge. The pixel samples in the picture area represented by a slice can be decoded without using any data from other slices. Flexible Macroblock Ordering (FMO) modifies the way the pictures are partitioned into slices and macroblocks by utilizing the concept of *slice groups*. Each *slice group* is a set of macroblocks defined by a macroblock to slice group map. This map consists on a slice group identification number for each macroblock in the picture, specifying which slice group

the associated macroblock belongs to. Regardless of whether FMO is in use or not, each slice can be coded using the different coding types:

- ***I-slice***: all macroblocks of the slice are coded using only intra prediction;
- ***P-slice***: in addition to the Intra coding types, the P-slice macroblocks may be coded using inter prediction with at most one motion-compensated prediction signal per partition block;
- ***B-slice***: in addition to the I and P coding types, B-slice macroblocks may be coded using inter prediction with two motion-compensated prediction signals per partition block.

- **Adaptive Frame/Field Coding Operation -** In interlaced frames with moving objects, regions or camera motion, two adjacent rows tend to show a reduced degree of statistical dependency when compared to progressive frames. In this case, it may be more efficient to compress each field separately. With this purpose in mind, H.264/AVC allows to use the following options when coding a frame:
  - ***Frame mode* -** the two fields are combined together to code them as one single coded frame;
  - ***Field mode* -** the two fields are not combined together and thus are coded as separate fields;
  - ***Picture-adaptive frame/field coding (PAFF) mode* –** an interlaced frame can be coded as a frame picture (i.e. the two fields are combined together and compressed as a single frame) or as two field pictures (top and bottom fields are coded separately) [1].

2.1.2.3. Main Novel Coding Tools

This section intends to present the new H.264/AVC coding tools that are mostly responsible for the compression gains regarding the previous standards.

- **Intra-Frame Prediction -** In all slice-coding types, Intra_4×4 , Intra_16×16 and I_PCM are the intra coding modes supported. Intra_4×4 is based on predicting each 4×4 luma block separately and it is typically used to predict regions that have significant detail. Intra_16×16 performs prediction for whole 16×16 luma blocks and it is typically more suited for coding very smooth areas. The I_PCM coding type allows the encoder to bypass the prediction and transform coding processes and instead directly send the sample values. Intra prediction in H.264/AVC is always conducted in the spatial domain and it may induce error propagation in environments with transmission errors since the intra prediction may be performed using neighboring inter-coded macroblocks [1].
- **Inter-Frame Prediction -** It is well know that the Inter prediction efficiency is critical for the overall RD performance; this was confirmed in the design of the H.264/AVC standard which significantly improved the temporal prediction process at the cost of additional complexity. Contrary to previous standards, the *B-slice* concept is generalized in H.264/AVC, e.g. other images can refer images containing *B-slices* for prediction with motion compensation.
  - ***Inter-frame prediction in P-slices***: Various predictive or motion compensated coding modes are specified in the P-slice coding. Each *P macroblock* mode corresponds to a specific partition of the macroblock into the block shapes used for motion-compensated prediction. The prediction signal for each predictive-coded $M \times N$ luma block is obtained by displacing a similar area in the corresponding reference picture, which is specified by a translational motion vector and a picture index (due to the available multiple

references). The motion compensation accuracy is in units of one quarter of pixel (this corresponds to the distance between luminance samples). The H.264/AVC syntax supports multipicture motion-compensated prediction as shown in Figure 4 [1]; the use of multiple references increases the RD performance, notably at the cost of memory and computational power.

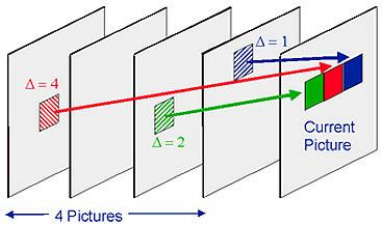


Figure 4 - Multi-frame motion compensation example for a P-slice [6].

- ***Inter-frame prediction in B-slices***: As it was referred, pictures containing *B-slices* can be used as reference for motion-compensated prediction, depending on the memory management control operation of the decoded picture buffer. The main difference between B and P-slices is that B-slices may include modes for which a prediction signal is a weighted average of two distinct motion-compensated predictions while P-slices, may only use one prediction reference. In B-slices, four different types of inter-picture prediction are supported: list 0, list 1, bi-predictive and direct prediction [1].

- **Transform, Scaling and Quantization:** As before, H.264/AVC transforms the temporal prediction residual to exploit the spatial redundancy after the temporal redundancy. However, in this standard, the transform is applied to 4×4 blocks and a separable integer transform with similar properties to a 4×4 DCT is used; later an alternative 8×8 DCT transform was also included. The Integer DCT transform matrix is given as:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

  As mentioned above, the Intra_16×16 mode and the chroma intra modes are intended for smoother areas. For that reason, the DC components undergo a second transform, obtaining transform coefficients that cover the whole macroblock, thus allowing to better exploit the spatial redundancy. A quantization parameter is used for determining the quantization level of the H.264/AVC transform coefficients. This parameter can take 52 values and these values are arranged so that an increase of 1 in the quantization parameter corresponds to an increase of approximately 12% of the quantization step size. This change of 12% in the quantization step size translates roughly into a bitrate reduction of approximately 12% [1] [5].

- **Entropy Coding:** In H.264/AVC, two entropy coding methods are supported. These are called *context-adaptive variable length coding* (CAVLC) and *context-adaptive binary arithmetic coding* (CABAC). CABAC has higher complexity than CAVLC but has better coding efficiency. When using CAVLC, the quantized transform coefficients are coded using VLC tables that are switched depending on the values of previous syntax elements. The efficiency can be further improved by using CABAC because it uses context-conditional probability estimates that are adapted to non-stationary statistical behaviors. Compared to

CAVLC, CABAC typically reduces the bitrate 10%-15% for the same quality at the cost of some additional complexity.

- **In-Loop Deblocking Filter:** One particular characteristic of block-based coding is the production of visible block structures, the so-called *block effect*. To attenuate this effect, H.264/AVC defines an adaptive in-loop deblocking filter where the strength of the filtering is controlled by several syntax elements [1]. The basic idea is that if a relatively large absolute difference between samples near a block edge is measured, it is probably a blocking artifact and should be filtered. However, if the magnitude of that difference is so large that it cannot be explained by the coarseness of the quantization used in the encoding process, the edge more likely expresses the actual characteristics of the source picture and should not be filtered. This filter reduces the bitrate by 5%-10% and an improved subjective video quality is visible; an example image is presented in Figure 5.

Figure 5 - Example of deblocking filter performance: without (left) and with (right) deblocking filter [7].

As the H.264/AVC standard includes many tools and addresses many applications with different functional needs, *profiles* and *levels* are specified to define conformance points to facilitate the interoperability while limiting the complexity (as less tools are included). A *profile* defines a set of tools that can be used to generate a conforming/compliant bitstream, whereas a *level* places constraints on certain key parameters of the bitstream such as the bitrate. There are at least seven profiles in H.264/AVC as shown in Figure 6:

- ***Baseline* profile** - Targets applications with low complexity and low delay requirements; it does not include several H.264/AVC tools, notably B-slices, CABAC and field coding.
- ***Extended* profile** - Hierarchical to the *Baseline* profile, it includes all Baseline tools plus several error resilience tools, notably data partitioning and SI/SP slices, but also B-slices.
- ***Main profile*** - This profile adds to the *Baseline* profile the B-slice, CABAC and field coding tools making it more efficient (and naturally complex).
- ***High* Profile** - Hierarchical to the *Main* profile but with the following enhancements: adaptive macroblock level switching between 4×4 and 8×8 transforms, quantization matrices defined by the encoder and independent encoder control of the quantization for each chrominance. This is the most efficient non-professional profile largely used in digital TV and Blu-ray.
- ***High 10* profile** - First professional profile, hierarchical to the *High* profile, notably adding the ability to encode videos with 9 and 10 bits per sample.

- ***High 4:2:2* profile** - Hierarchical to the *High 10* profile, adding the ability to encode the 4:2:2 subsampling format.
- ***High 4:4:4* profile** - Hierarchical to the *High 4:2:2* profile, adding the abilities to encode samples with 11 and 12 bits, in the 4:4:4 subsampling format, the color residual transform and lossless predictive coding [5].

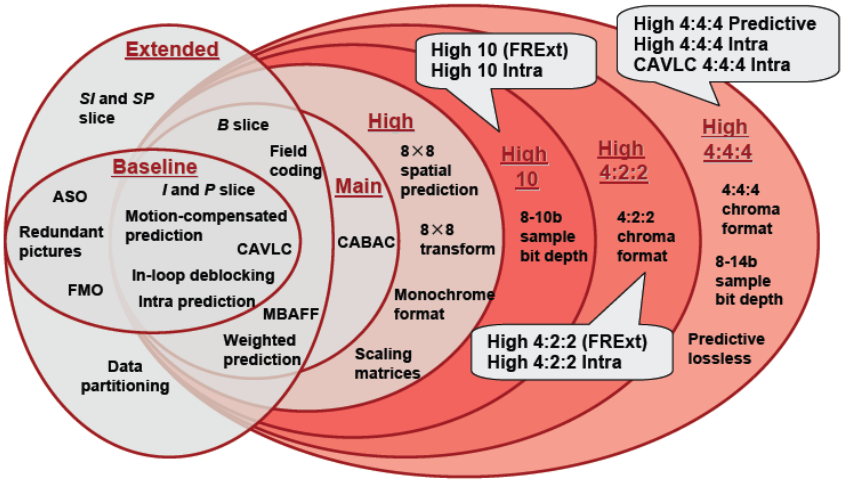


Figure 6 - Graphical representation of the H.264/AVC profiles [8].

2.1.2.4. Performance Assessment

To illustrate the H.264/AVC RD performance, some experimental results for entertainment-quality applications (DVD video systems and HDTV) are presented in the following. Video sequences in such applications are usually encoded at resolutions of 720×480 pixels and higher, at average bitrates of 3 Mbit/s and up. The MPEG-2 Video standard was used for comparison purposes [7]. Figure 7 illustrates the RD performance – MPEG-2 Video Main and H.264/AVC Main profiles - and bitrate savings curves for a typical video entertainment standard definition (SD) sequence. The curves demonstrate that H.264/AVC offers significant rate savings that lie between 45% and 65% at lower bitrates, and between 25% and 45%, for the higher bitrates [7].

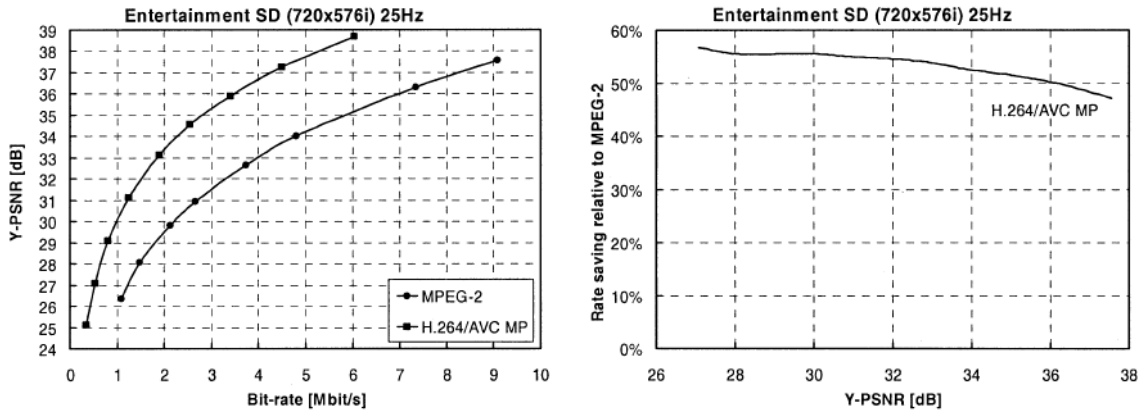


Figure 7 – RD performance and bitrate savings plot for entertainment-quality applications [7]

In general, the H.264/AVC brings about 50% compression gains regarding the best previously available standards for the various applications and conditions. As encoders are not normative, naturally these gains imply the usage of efficient encoder control methods.

2.2. Reviewing Distributed Video Coding

This section intends to briefly overview the Distributed Video Coding (DVC) paradigm which is also important for this Thesis. In this context, some basic aspects will be presented and early Wiener-Ziv video coding architectures will be mentioned. Finally, the DISCOVER DVC codec will be introduced as one of the best representatives of the DVC technology.

2.2.1. Basic Concepts and Early Wyner-Ziv Video Coding Solutions

To address the needs of some emerging applications, around 2002, some research groups revisited the video coding problem at the light of two Information Theory results already from the seventies: the Slepian-Wolf and the Wyner-Ziv theorems. These efforts originated the DVC paradigm and also Wyner-Ziv (WZ) video coding, as a particular DVC case. The Slepian-Wolf theorem addresses the case where two statistically dependent discrete random sequences, independently and identically distributed (i.i.d.), X and Y, are independently encoded, and thus not jointly encoded as in the largely deployed predictive coding solutions. The Slepian-Wolf theorem states that the minimum rate to encode the two sources is the same as the minimum rate for joint encoding, with an arbitrarily small error probability. In theory, the rate bounds for a vanishing error probability considering two sources are:

$$R_x \geq H(X|Y) \tag{1}$$

$$R_y \geq H(Y|X) \tag{2}$$

$$(R_x + R_y) \geq H(X,Y) \tag{3}$$

This means that the minimum coding rate for distributed coding is the same as for joint encoding (i.e. the joint entropy), provided that the individual rates for both sources are higher than (or equal to) the respective conditional entropies. Later, the Wyner-Ziv theorem states that when performing independent encoding with side information, there is no coding efficiency loss under certain conditions with respect to the joint encoding case, even if the coding process is lossy (and not anymore asymptotically lossless as in the Slepian-Wolf case).

Based on these two theorems, a new video coding paradigm known as Distributed Video Coding has emerged. DVC does not rely on joint encoding, and thus, when applied to video coding, it results on the absence of the temporal prediction loop and lower complexity encoders. In this context, DVC may provide the following functional benefits: flexible encoder/decoder allocation of the global video codec complexity, improved error resilience, codec independent scalability and exploitation of multiview correlation without cameras/encoders communicating among them [9]. The first practical WZ video coding solutions emerged around 2002 at Stanford University and at the University of California, Berkeley. The Stanford WZ architecture is mainly characterized by frame-based Slepian-Wolf coding, typically using turbo codes, and a feedback channel to perform rate control at the decoder. The Berkeley WZ video coding solution, well known as Power-efficient, Robust, high-compression, Syndrome-based Multimedia (PRISM), is mainly characterized by block-based decoder motion estimation. For a more detailed description of these solutions, please refer to [9]. From a technical point of view, the main

functional differences between the two early WZ video codecs can be stated as follows (Stanford vs Berkeley) [9]:

1. Frame-based versus block-based coding: in the latter approach, it is easier to accommodate coding adaptability to address the highly non-stationary statistics of video signals;
2. Decoder rate control versus encoder rate control: in the former case, a feedback channel is needed, restricting the scope to real-time applications, but making the rate control problem much simpler;
3. Simple encoder versus smarter and more complex encoder: the latter case allows incorporating spatially varying coding mode decisions;
4. More sophisticated channel codes versus simpler channel codes;
5. No auxiliary data transmitted versus hash codes sent by the encoder to help the decoder in the motion estimation process;
6. Less intrinsically robust to error corruption versus higher resilience to error corruption due to the PRISM motion search like approach performed at the decoder, which allows finding better side information and thus reducing the residual noise.

In recent years, with the extensive research on DVC, many developments have been introduced in both early WZ codecs, with the consequence of significantly improving its RD performance. Since Slepian-Wolf coding is the core of WZ coding, and channel coding plays a central role in Slepian-Wolf coding, channel coding developments play an important role not only in terms of RD performance but also in terms of codec complexity budget. Another key improvement has been the enhancement of the side information (a decoder estimation of the original frame to code) quality as this quality plays a central role in the WZ codec´s overall RD performance. Without a powerful side information creation mechanism, no competitive RD performance can be achieved. Regarding correlation noise modeling, since WZ video coding targets the lossy coding of the difference between the original data and its corresponding side information, it is essential for an efficient RD performance that the decoder (and sometimes the encoder) are aware of the statistical correlation between the original and side information. Somehow inspired by the PRISM approach, the addition of a block classification module to the Stanford-based WZ video architecture has been proposed, allowing the selection of one of two coding modes (Intra or WZ modes), depending on the available temporal correlation. This approach results from the observation that the correlation noise statistics describing the relationship between the original frame and its corresponding side information available at the decoder is not spatially stationary. Last, regarding the side information creation issue, it was found that a way to overcome the “blind” frame-based SI creation approach adopted by the early Stanford WZ video coding solution was for the encoder to have the capability to send some hash signatures to help the decoder to generate better side information, notably for the most critical areas/blocks. Since the hash requires fewer bits than the original data, the encoder is allowed to keep the hash codewords from the previous frame in a small hash store. Significant gains over conventional DCT-based Intra frame coding were reported with comparable encoding complexity.

For several reasons, the Stanford DVC architecture has been adopted more often by the research community, notably by the IST Multimedia Signal Processing group; for this reason, one of the best representatives of the Stanford DVC approach will be briefly described in the following: the DISCOVER project WZ codec.

2.2.2. The DISCOVER Wyner-Ziv Video Codec

This section intends to provide a brief overview of the DISCOVER WZ video codec since it is considered one of the most advanced and best performing WZ codecs.

2.2.2.1. Architecture and Walkthrough

The DISCOVER WZ architecture, including encoder and decoder, is presented in Figure 8.

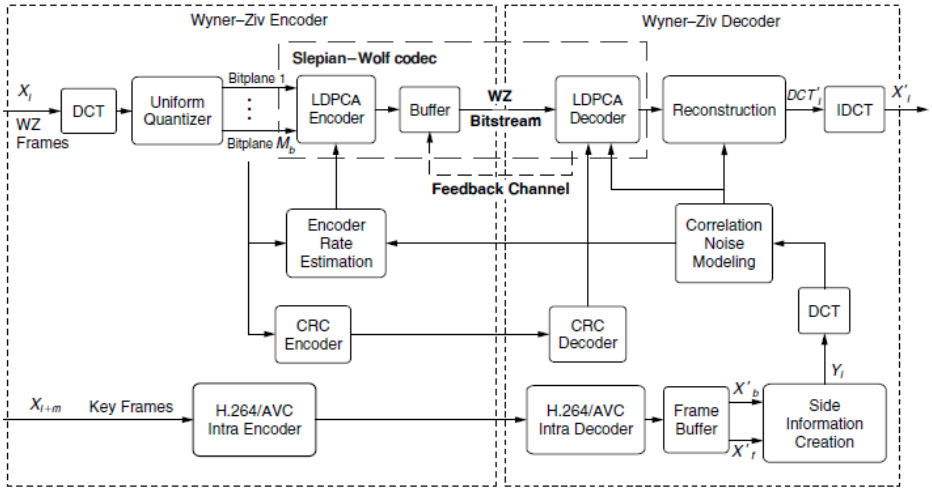


Figure 8 - DISCOVER Wyner–Ziv video codec architecture [9].

To better understand the DISCOVER WZ codec processing chain, both the encoder and decoder walkthroughs are presented in the following [9].

At the encoder, the following steps are performed:

1. **Frame Classification:** First, a video sequence is divided into *WZ frames*, this means the frames that will be coded using a WZ coding approach, and the so-called *key frames* that will be conventionally coded as Intra frames; these frames are used by the decoder to create the *side information* frame. Key frames are periodically inserted with a certain *Group of Pictures* (GOP) size. An adaptive GOP size selection process may also be used but most results available in literature use a GOP size of 2, which means that odd and even frames are key frames and WZ frames, respectively.
2. **Discrete Cosine Transform:** Over each WZ frame, an integer 4×4 block-based DCT is applied. The DCT coefficients of the entire WZ frame are then grouped together according to the position occupied by each DCT coefficient within the 4×4 blocks, forming the so-called *DCT coefficient bands* from the DC band to the highest frequency band.
3. **Quantization:** Each DCT coefficient band b_k is uniformly quantized with 2^{M_k} levels and after, bit-plane extraction is performed over the resulting quantized symbol stream. For a given band, the quantized symbols bits of the same significance are grouped together forming the corresponding bit-plane array, which is then independently low-density parity-check (LDPC) encoded.
4. **LDPC Encoding:** The LDPC procedure starts with the most significant bit-plane array, which corresponds to the most significant bits of the b_k band quantized symbols. The parity information is stored in the buffer and sent in chunks upon decoder request.

5. **Encoder Rate Estimation:** To limit the decoding complexity and transmission delay, the encoder estimates for each bit-plane the initial number of bits to be sent before any request is made. If the rate is underestimated, the decoder will complement it by making one or more requests via a feedback channel for additional information.

At the decoder, the following steps are performed:

1. **Side Information Creation:** The decoder creates the side information (SI) for each WZ frame corresponding to an estimation of the original WZ frame based on the previously decoded frames; the better the quality of the estimation, the smaller are the number of errors the WZ LDPC decoder has to correct and, thus, the bitrate necessary for successful decoding.
2. **DCT Estimation:** A block based 4×4 DCT is carried out over the side information to obtain the SI DCT coefficients, which are the estimates of the corresponding WZ frame DCT coefficients under decoding.
3. **Correlation Noise Modeling:** The residual statistics between corresponding WZ frame DCT coefficients and the side information DCT coefficients are assumed to be modeled by a Laplacian distribution. The distribution parameters have to be estimated for the next channel decoding step.
4. **LDPC Decoding:** Once the DCT-transformed side information and the residual statistics for a given DCT coefficients band b_k are known, the decoded quantized symbol stream associated with the DCT band b_k can be obtained through a LDPC decoding procedure.
5. **Request Stopping Checking:** To decide if more bits are necessary to decode a certain bit-plane, a request stopping criterion is checked, notably by determining if all LDPC parity-check equations are fulfilled for the decoded codeword.
6. **CRC Checking:** Because some residual errors may be left even when all LDPC parity-check equations are fulfilled, a CRC checksum is transmitted to help the decoder to detect and correct the remaining errors in each bit-plane.
7. **Further LDPC Decoding:** After successfully LDCPA decoding the most significant bit-plane array of the b_k band, the LDPC decoder proceeds in an analogous way to the remaining M_{k-1} bit planes associated with that band. This procedure is repeated until all the DCT coefficients bands for which WZ bits are transmitted are LDPC decoded.
8. **Symbol Assembling:** After LDPC (or turbo) decoding the M_k bit-planes associated with the DCT band b_k, the bit-planes are grouped together to form the decoded quantized symbol stream associated with the b_k band. This procedure is performed over all the DCT coefficients bands to which WZ bits are transmitted.
9. **Reconstruction:** Once all quantized symbol streams are obtained, the matrix with the decoded DCT coefficients for each block can be reconstructed using an appropriate inverse quantization tool.
10. **IDCT:** After, a block-based 4×4 IDCT is performed, and thus the reconstructed pixel domain WZ frame is obtained.
11. **Frame Remixing:** To finally get the decoded video sequence, decoded key frames and WZ frames are mixed in the appropriate way.

2.2.2.2. Performance Assessment

This section will present some performance results for the DISCOVER WZ video codec, notably to assess the performance gap regarding predictive video coding here represented by

appropriate standard configurations. Only the luminance component is coded and thus all the metrics in this section refer only to the luminance. All the frames were used for each video sequence, coded at Quarter Common Intermediate Format (QCIF) spatial resolution, 15Hz and GOP sizes of 2, 4 and 8 [9].

The RD performance regards how the overall rate for the key frames and the WZ frames translates into quality. The used quality metric is the PSNR over all the frames of the video sequence coded with a certain quantization matrix. The standard coding solutions used for comparison purposes are H.263+Intra, H.264/AVC Intra and H.264/AVC Inter No Motion (with temporal prediction but no motion estimation/compensation) as they should have all low encoder complexity since no motion estimation is performed at the encoder as for the DISCOVER WZ codec. The RD performance results are presented in Figure 9 and Figure 10. From these results, the following conclusions can be drawn: for the *Coast Guard* and *Hall Monitor* sequences, there are coding gains for the DISCOVER DVC codec for all RD points and all GOP sizes, with reported average gains up to 9 dB when compared with H.263+ Intra. Knowing that H.263+ Intra does not exploit the temporal redundancy, these results show that the DISCOVER WZ codec can exploit, at least partly, the temporal correlation in the video content. For content with high and medium motion, when the key frames are separated by a longer gap in time, the side information quality decreases (and thus the overall RD performance) since it becomes more difficult to estimate the side information for the frames in between. It is visible that complex and erratic motion causes a poorer RD performance, especially when the GOP size is larger. For all sequences but *Hall Monitor*, the DISCOVER WZ codec with GOP size 2 wins regarding other GOP sizes, showing the difficulty is getting good side information for longer GOP sizes due to a decrease in the performance of the frame interpolation (FI) process used for the side information creation. In terms of RD performance, the DISCOVER codec has significant advantages when compared with standard solutions such as H.264/AVC Intra. For low motion sequences, it is even possible for DISCOVER to have better results against the H.264/AVC No Motion codec. For the cases with longer GOP sizes, it is more difficult to outperform H.264/AVC Intra due to difficulty in getting good side information, notably when the key frames are farther away.

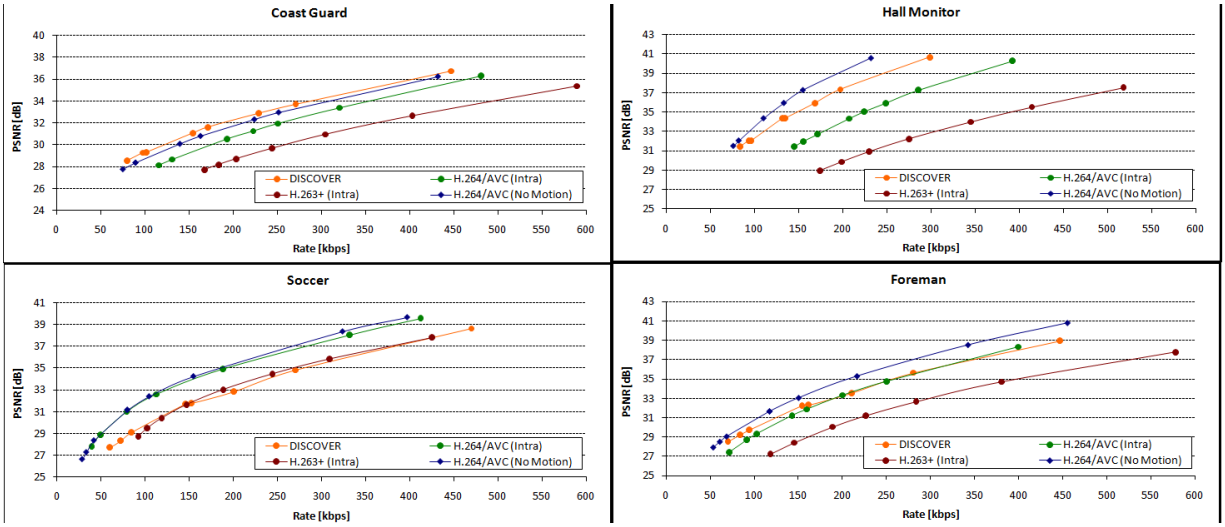


Figure 9 - RD performance comparison for GOP size 2 [10].

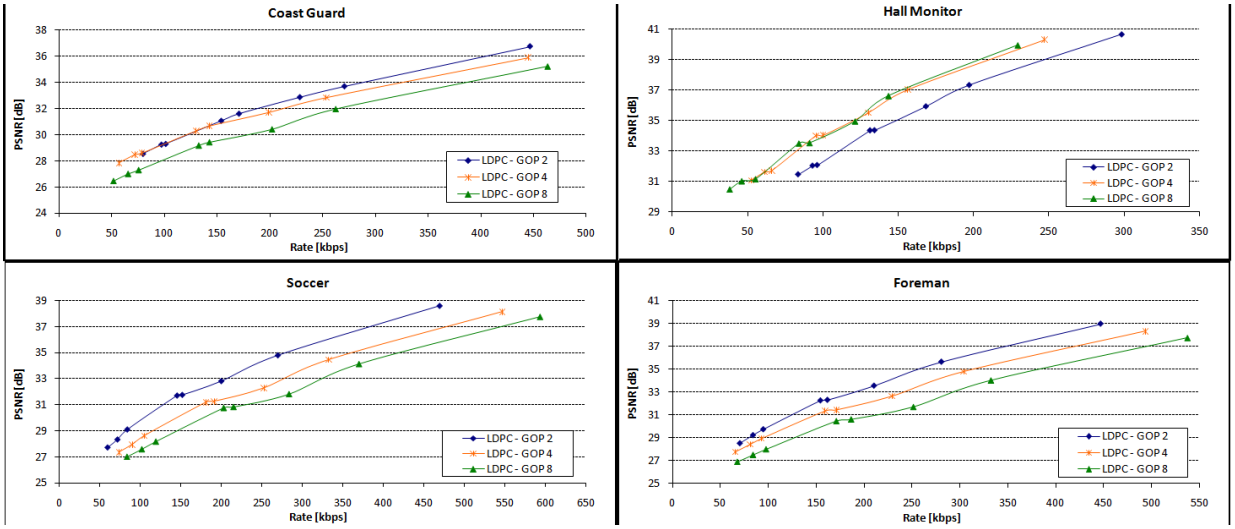


Figure 10 - RD performance comparison for GOP sizes 2, 4 and 8 [10].

Regarding the complexity assessment, it is important to acknowledge that the encoding process includes two components: the WZ frames encoding and the key frames encoding. The larger the GOP size, the smaller the number of key frames coded and, thus, the lower will be the share of the key frames in the overall complexity. With the complexity results available in [9] it is possible to conclude that for the DISCOVER WZ codec the WZ frames encoding complexity is small when compared to the key frames encoding. In fact, the DISCOVER WZ codec encoding complexity is lower than the H.264/AVC Intra and H.264/AVC No Motion encoding complexities and it decreases with the GOP size. If the encoding complexity is a critical requirement for some application, the complexity assessment results in [9] together with the RD performance previously shown indicate that the DISCOVER WZ video codec with GOP size 2 is already a credible solution regarding the conventional coding alternatives [9]. However, the situation is different for the decoding complexity that also includes two major components: the WZ frames decoding and the key frames decoding. The decoding complexity results available in [9] it is possible to conclude that for the DISCOVER WZ codec the key frames decoding complexity is negligible regarding the WZ frames decoding complexity. Thus, contrary to the encoding complexity, the longer the GOP size, the higher is the overall decoding complexity, since the higher is the share of WZ frames. The WZ decoding complexity increases significantly when the bitrate increases since the number of bit-planes to LDPC decode is higher. The LDPC decoding is the main responsible for the higher decoding complexity as it works iteratively for each bitrate request until all bit-planes of all DCT bands are decoded.

2.3. Reviewing Relevant Background on Quantization

Considering the relevance of this topic for the Thesis, this section will present some relevant background information on quantization, from the basic concepts to the specific aspects of H.264/AVC quantization and a couple of adaptive (encoder) quantization solutions.

2.3.1. Basic Issues on Scalar Quantization

Quantization can be described as a process where the continuous range of values of a sampled input signal is divided into non-overlapping sub-ranges, and to each sub-range a discrete output value is assigned. The operation of a quantizer involves the application of a classification rule to produce quantization indices and then applying entropy coding to the

quantization indices to transmit them through a communication channel to the decoder. Thus, it is possible to represent the quantized signal with fewer bits than the original signal introducing, however, some quantization error. The quantization process is lossy because the visual irrelevance of the signal may be exploited to reduce the bitrate necessary to code the video signal at the cost of an additional quantization error which may (or may not) be perceptually irrelevant.

In image and video compression codecs, the quantization process is usually made in two parts: a forward quantizer (FQ) in the encoder and an inverse quantizer (IQ) in the decoder. A critical parameter is the quantization step size between re-scaled values that defines how coarse the signal is represented. A larger step size translates into a higher compression ratio, but also a cruder representation of the original signal. On the other hand, with a smaller step size, the quantized values are more approximately matched with the original ones but the compression ratio is smaller [11].

In image and video coding, quantization reduces the precision of the DCT coefficients by removing the ones that are insignificant such as near-zero coefficients and quantizing the non-zero coefficients. The forward quantizer is designed to map insignificant coefficient values to zero whilst retaining the significant non-zero coefficients; its typical output is an array of quantized coefficients. A scalar quantizer can be decomposed into a function $C(x)$ called a *classification rule* that selects an integer-valued class identifier called the *quantization index;* this function is performed at the encoder. A second function, $R(k)$, called a *reconstruction rule* produces a real valued decoded output $Q(x) = R[C(x)]$ called a *reconstruction value;* this function is performed at the decoder side. A well known but rather simple quantizer reconstruction rule is the so-called *nearly-uniform-reconstruction quantizer* (NURQ). The reconstruction rule for a NURQ uses two parameters, a step size, *s,* and a non-zero offset parameter, *p,* and is defined as:

$$R(k) = sign(k) \times s \times (|k| + p) \tag{4}$$

where $sign(k)$ is a function equal to 1 when $k > 0$, equal to $0\ if\ k = 0$ and equal to $-1\ if\ k < 0$. While typically $p \geq 0$, an important NURQ case is the *uniform reconstruction quantizer (URQ)*, which is defined as a NURQ with $p = 0$. This reconstruction rule is important as it was adopted by the H.264/AVC video coding standard. Another important case is $p = 1/2$ due to its prevalence in previous standards for image and video coding. It is important to notice that although the *reconstruction function* is normative, the *classification rule* is not, which allows to perform optimizations suitable to the data being coded as long as the normative reconstruction function is known.

The classification region corresponding to $C(x)$ equal to zero is called the *dead-zone*. One effective quantization classification rule for the NURQ is the so-called *dead-zone plus uniform threshold quantization* (DZ+UTQ) solution. This rule works as follows [12]:

$$C(x) = sign(x) \times max\left(0, floor\left(\frac{|x|}{s} + 1 - p - z\right)\right) \tag{5}$$

where the additional parameter, *z,* known as *rounding offset,* controls the width of the *dead-zone*, which is equal to $2s(p + z)$, and the function *floor(.)* is defined as the largest integer less than or equal to its argument. A simple example of a quantization structure is presented in

Figure 11, where the crosses on the number line indicate the location of the NURQ reconstruction values and the solid vertical lines indicate the threshold values that form the decision regions.

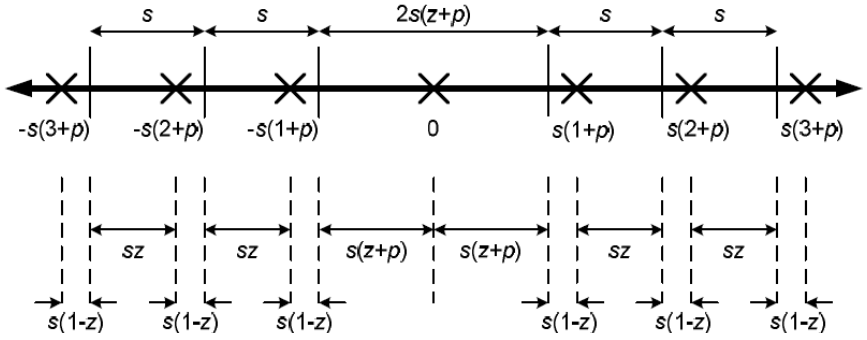


Figure 11 – Example of a quantizer structure [12].

2.3.2. Transform and Quantization in H.264/AVC

H.264/AVC is fundamentally a lossy compression format, in which a degree of visual distortion is introduced into the video signal as a trade-off with lower rate and thus higher compression performance. Compression efficiency is the ultimate reason for introducing a transform. As H.264/AVC heavily relies on efficient prediction before the transform, with the use of 4×4 Intra modes (spatial prediction) and Inter modes (temporal prediction) significantly reducing the correlation between neighboring 4×4 blocks, a smaller 4×4 transform support was selected [13]. After prediction, transform and quantization, the video signal is represented as a series of quantized transforms coefficients together with some auxiliary prediction parameters that are coded into a bitstream, e.g. motion vectors and coding modes. H.264/AVC provides several mechanisms for converting symbols and parameters into a compressed bitstream, namely fixed length binary codes, variable length Exponential-Golomb codes, CAVLC and CABAC.

In H.264/AVC, the transform and quantization processes are designed to minimize the computational complexity and to avoid encoder/decoder mismatch. This is achieved both by using a core transform that can be carried out using integer or fixed-point arithmetic and by integrating a normalization step with the quantization process to minimize the number of multiplications required to process a block of residual data. The scaling and inverse transform processes carried out by a decoder are exactly specified in the standard so that every H.264/AVC implementation produces identical results [11]. H.264/AVC makes extensive use of prediction tools since even the intra coding modes rely on spatial prediction; as a consequence, H.264/AVC is very sensitive to prediction drift. As an example, in an I-frame, 4×4 blocks can be predicted from their neighbors and thus, at each stage, prediction drift can accumulate. For a *Common Intermediate Format* (CIF) image with a size of 88 4×4 blocks in a row, prediction drift can accumulate 88 times while decoding one I-frame row of blocks. Thus, it becomes clear that, as a result of the extensive use of prediction in H.264/AVC, the residual must be drift-free. In addition H.264/AVC have minimized the complexity of some of its tools, for example, some criteria was developed to restrict the complexity of the inverse transform, such as using only 16 bit multiplications and 16 bit memory access. As for past standards, an early H.264/AVC design feature was the variation of the quantization step size which increases by approximately 12% for each increase in the quantization parameter, so that each increment of six in the

quantization parameter doubles the quantization step size. To better understand the role of the quantization step and the quantization parameter in H.264/AVC, the following equations were established. In H.264/AVC, the quantized coefficients may be obtained by a simple division as follows or some other similar classification rule:

$$C^Q = round\left(\frac{C}{Q_{step}}\right) \tag{6}$$

where C^Q are the quantized parameters and Q_{step} is the chosen quantization step size. In H.264/AVC, the quantization step is related to the quantization parameters according to the following formula:

$$Q_{step}(QP) = Q_{step}(QP\%6) \cdot 2^{floor\left(\frac{QP}{6}\right)} \tag{7}$$

where QP is the quantization parameter and $x\%y$ defines the remainder of the division of x by y [11]. For the range of typical values for *QP* and Q_{step}, please report to the tables available in [11].

2.3.2.1. H.264/AVC Integer Transform Design

The DCT is commonly used in block based transform coding and it maps a length-N vector *x* into another vector *X* of transform coefficients by a linear transformation $X = Hx$ where the element in the k^{th} row and the n^{th} column of H is defined by:

$$H_{kn} = H(k,n) = c_k\sqrt{\frac{2}{n}}cos\left[(n+\frac{1}{2})\frac{k\pi}{N}\right] \tag{8}$$

for the frequency index $k = 0,1,\dots,N-1$ and the sample index $n = 0,1,\dots,N-1$ with $c_0 = \sqrt{2}$ and $c_k = 1$ for $k > 0$. As H are irrational numbers, it may not be possible to obtain $u(n) = x(n)$ for all the *n* when computing $X = Hx$ and $u = round\{H^T X\}$, if the direct and inverse transforms are implemented in different machines with different floating point representations and rounding. If appropriate scale factors are introduced such that $X = round\{\gamma Hx\}$ and $u = round\{\beta H^T \mathrm{X}\}$, then it is possible to make $u(n) = Gx(n)$ where G is an integer for almost all *n* by choosing β large enough and γ appropriately. Still, it is not possible to guarantee exact results unless the full intermediate rounding procedures are standardized. Thus, it is beneficial to replace *H* by an orthogonal matrix with integer entries.

There are two basic approaches that can be used for that purpose: one is to build *H* with just a few integers with symmetries similar to those of the DCT, which guarantees orthogonality. The matrix *H* is as follows:

$$H = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \tag{9}$$

The original H.264/AVC design chose $a = 13$, $b = 17$ and $c = 7$ which made *H* close to a scaled DCT and then ensured that all rows had the same norm. Another approach is to round the scaled entries of the DCT matrix to the nearest integers as follows:

$$H = round\,\{\alpha H_{DCT}\} \tag{10}$$

where H_{DCT} is the DCT matrix and α is an adjustable parameter. The problem with the choice of $a = 13$, $b = 17$ and $c = 7$ is that it increases the dynamic range. If $\max\{|x(n)|\} = A$ then $\max\{|X(k)|\} = 52$, which means that the transform has a dynamic range gain of 52. So the total gain is $52^2 = 2704$ and, as $log_2(2704) = 11.4$, twelve more bits are needed to store $X(k)$ than to store $x(n)$.

To overcome this limitation, Malvar et al. proposed in [13] to set $\alpha = 2.5$ in (10) which translated into the following matrix:

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \tag{11}$$

This way as the maximum sum of absolute values in any row equals 6, the maximum dynamic range gain increase for the transform is $log_2(6^2) = 5.17$ which means that the storage of $X(k)$ needs only six more bits than $x(n)$.

At the decoder, it is possible to use the transpose of *H* in (9) as long as the reconstructed transform coefficients are scaled properly. However, to minimize the combined rounding errors from the inverse transform and reconstruction, the dynamic range gain has to be reduced. A possible solution is to scale the odd-symmetric basis functions by $\frac{1}{2}$ in (9). In this way, the sum of absolute values for the odd functions is cut in half to 3 and the maximum sum of absolute values for any basis function now equals 4, thus reducing the dynamic range gain for the 2-D inverse transform from 6^2 to 4^2; as $log_2(4^2) = 4$, the increase in dynamic range is reduced from 6 to 4 bits. The inverse transform matrix is then defined by:

$$\breve{H}_{inv} = \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix} \tag{12}$$

where $\breve{H}_{inv}$ is a scaled inverse of *H* [13].

2.3.2.2. H.264/AVC Quantization Process

For a given step size, Q_s, usually an integer, the encoder can perform quantization by:

$$X_q(i,j) = sign\{X(i,j)\}\frac{|X(i,j)| + f(Q_s)}{Q_s} \tag{13}$$

where *i* and *j* are the row and column indices and $f(Q_s)$ controls the quantization width near the origin, well know as the *dead-zone* as mentioned before. The decoder can perform inverse quantization also called *reconstruction* by scaling the quantized data by Q_s:

$$X_r(i,j) = Q_s X_q(i,j) \tag{14}$$

This means in practice that H.264/AVC adopts a URQ uniform reconstruction quantizer corresponding to a NURQ with *p* equal to zero. The dead-zone control parameter, *f*, may be different for different encoders as they are not normative but it is typically in the range 0 to $1/2$.

To avoid divisions, thus reducing the decoder complexity, the formulas above are replaced by:

$$X_q(i,j) = sign\{X(i,j)\}[(|X(i,j)|A(Q) + f2^L) \gg L] \tag{15}$$

$$X_r(i,j) = X_q(i,j)\, B(Q) \tag{16}$$

$$x_r = (H^T X_r + 2^{N-1} e) \gg N \tag{17}$$

where $e = [1\ 1\ 1\ 1\ ]^T$, the new parameter *Q* varies from zero to Q_{max}, and the association of quantization parameters $A(Q)$ and $B(Q)$ is such that zero corresponds to the finest quantization and Q_{max} to the coarsest quantization. The complexity can be reduced even further by using formulas that allow for 16-bit arithmetic precision, with no penalty in PSNR performance. To achieve this goal, reduced values of $B(Q)$ and of the parameters *L* and *N* are used.

Another aspect in the original H.264/AVC quantization design in (16) is that the values of $B(Q)$ increase in approximately equal steps in an exponential scale, roughly doubling for every increase of six in *Q*. By forcing $B(Q)$ to double for every increase of 6 in *Q,* the size of the quantization and reconstruction tables can be reduced. Thus, the H.264/AVC solution [13] adopts the following quantization formula:

$$X_Q(i,j) = sign\{X(i,j)\}[(|X(i,j)|A(Q_M,i,j) + f2^{15+Q_E} \gg (15 + Q_E)] \tag{18}$$

where $Q_M \equiv Q \mod 6$ and $Q_E \equiv \frac{Q}{6}$. For every increase of one in the exponent Q_E, the denominator in (18) doubles, with no changes in the scaling factor multiplying $|X(i,j)|$. This periodicity enables a large range of quantization parameters without increasing the memory requirements. Although it is not completely clear in the reference paper, the fixed rounding offset seems to be integrated in the $A(Q)$ parameter.

The H.264/AVC inverse transform specification covers some additional aspects. Since for image regions with mostly flat pixel values there is significant correlation among transform DC coefficients of neighboring blocks, DC coefficients are grouped in blocks of size 4×4 for the luminance channel and blocks of size 2×2 chrominance channels and an additional transform is employed. This two level transform is usually referred as a *hierarchical transform.* In the original H.264/AVC design, the second level 4×4 transform was the same as the first level transform. However, in the final standard, a *Hadamard* transform (specified with $a = b = c = 1$ in (9)) is used because there were no reported performance losses in the standard video tests.

As H.264/AVC encoders are not normative while decoders are, the quantization process is not fixed while the inverse quantization process is. This implies the encoder has the freedom to optimize the quantization process for specific content based on the knowledge on the

standardized inverse quantization process. Relevant adaptive quantization solutions in the literature are presented in the next section.

2.3.3. Adaptive Quantization Algorithms

In this subsection, two adaptive quantization algorithms available in the literature are presented. They propose different approaches to adapt the rounding offset used in the encoding quantization process.

2.3.3.1. Adaptive Quantization using an Equal Expected-Value Rule

This method proposed by Sullivan in 2005 [12] is based on adjusting the rounding offset to maintain an equal expected value for the absolute value of the quantized data at the input and output of the quantization process. This adaptive quantization using an equal expected-value rule is able to provide up to 1 dB of improvement in RD performance for high decoded quality, i.e. high PSNR values.

A. The Quantization Process

The technique proposed by Sullivan in [12] adopts a DZ+UTQ quantization method and optimizes the quantizer performance for the normative H.264/AVC inverse quantization with a mean squared error distortion metric. One way to design a DZ+UTQ classification rule is to select the rounding offset, z, such that the mean of the absolute value of the input random variable |X| is equal to the mean of the absolute value of its reconstructed value:

$$E\{|Q(X)|\} = E\{Q(|X|)\} = E\{|X|\} \tag{19}$$

If the reconstruction value for every classification region of the quantizer is mean-squared optimal for this classification rule, the quantizer will also have this property. The mean estimator can be formed using a simple geometrically-decaying weighted sum of the sequence of samples, Y_i. Such sequence of estimates, M_i, can be formed by setting M_0 equal to some *a priori* guess for the actual mean, and then forming subsequent estimates as follows:

$$M_{i+1} = M_i + w_i \times (Y_i - M_i) \tag{20}$$

where $0 \leq w_i \leq 1$ is a weighting factor.

In this particular design problem, z should be in the range of $\frac{1}{2} \leq z \leq 1$, so a rough estimate in that range is used as the initial estimator, denoted as z_0. The value of z is then updated adaptively as the quantization process operates on the sequence of random input values. The samples falling in the *dead-zone* can also be separated from the method of selecting the optimal value of z, because if it is assumed that the input power density function is symmetric about zero, then the value of *z* does not affect the optimality of the *dead-zone* reconstruction value of zero. This leads to an encoding rule for the operation of the quantization process for a

sequence of input random variables X_i using a sequence of different values of z that are equal to random variables Z_i so that $z_i = Z_i$ is the threshold used in the classification rule for the quantization of X_i. For computing the quantization thresholds at the encoder, the following applies:

$$Z_{i+1} = Z_i + w_i \times \frac{I(Q(X) \neq 0) \times (|Q(X)| - |X_i|)}{s} \tag{21}$$

where $I(\cdot)$ is the indicator function defined to be 1 when its argument is true and 0 when its argument is false.

For best performance, each distinct source and each distinct type of quantizer should have its own adaptive rounding offsets. During the encoding, the rounding offset parameter is updated as follows (by using $f = 1 - z$ in (21)):

$$f_{i+1} = f_i + w_i \times I(Q[X] \neq 0) \times \frac{|X_i| - |Q[X_i]|}{s} \tag{22}$$

where $0 \leq w_i \leq 1$. A key issue is what should happen when the step size for quantization changes, for example due to the rate control operated during the H.264/AVC encoding. Ideally, the selected offset value, *z*, should depend on the step size in use; however, when the value of *s* is relatively stable, the method described for determining *z* may be sufficient for practical purposes.

B. Performance Assessment

The adaptive quantization encoding rule presented above was tested with a JM8.6 encoder with a rounding offset set as in (22). Figure 12 shows the RD curves for an encoder using the adaptive quantization rounding technique presented, and the JM8.6 reference software encoder, with 16 frequency components for luminance blocks in Intra 4×4 modes and 15 AC components for chroma 4×4 blocks in Intra mode. The experiments were conducted using several CIF and 4CIF format sequences with 300 frames per sequence and a frame rate of 30 frames per second. For a more accurate description of the test conditions and the encoder configurations, please refer to [12]. The results show up to 1 dB performance improvements (particularly at high) bitrates when using the discussed adaptive rounding method.

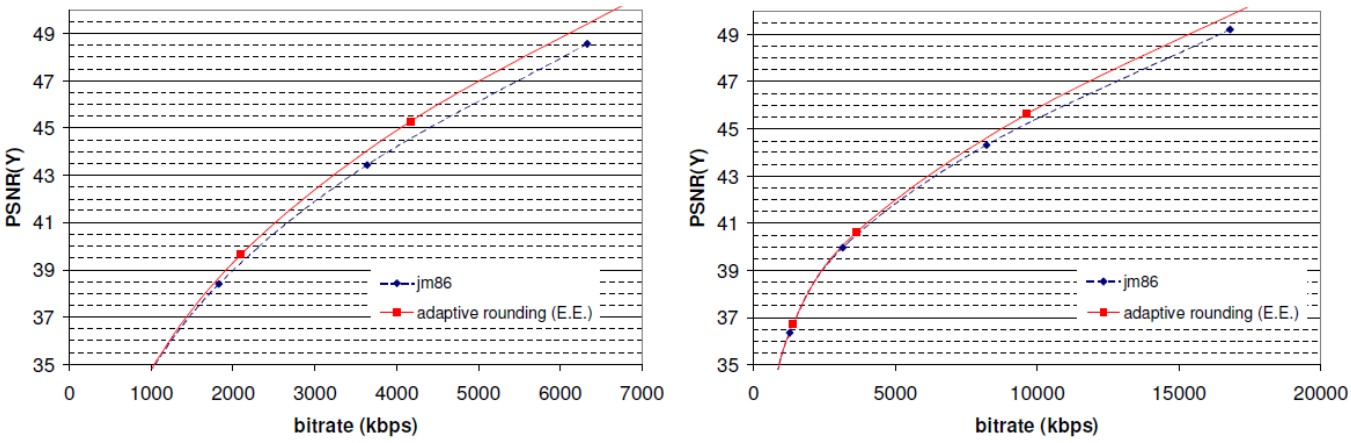


Figure 12 - Performance comparison between encoders with and without adaptive rounding method for the sequence Mobile CIF (left) and Soccer 4CIF (right) [12].

2.3.3.2. Adaptive Quantization based on Rounding Offsets[1]

This method, proposed by Xu and others [14], is known as Adaptive Rounding Offsets (ARO) and jointly adjusts the QP (a set of predefined values for the quantization step size, *q*), and the rounding offset, *s*, to reach high bitrate control accuracy. Unlike *QP,* which has a limited number of choices, *s* is a continuously adjustable variable and thus it may enable the rate control algorithm to reach any precision. More importantly, a linear rate model is proposed where $ln(R)$, with *R* being the bitrate, is related to *s* in a linear fashion for a given *QP*. For a detailed review of the linear relationship between $ln(R)$ and *s,* please refer to [14].

A. The Quantization Process

In recent video coding standards, the rounding offset, *s*, together with the quantization step size, *q*, are used to quantize the transformed coefficients, *W*. In H.264/AVC encoding, *W* is typically quantized as:

$$Z = \left\lfloor \frac{|W|}{q} + s \right\rfloor \cdot sgn(W) \tag{23}$$

where *Z* is the quantization level of *W*. The function $\lfloor \cdot \rfloor$ rounds a value to the nearest integer that is less than or equal to its argument, while $sgn(\cdot)$ returns the sign of the input signal. If a quantization matrix is used, *W* is scaled with the corresponding matrix element before quantization. At the H.264/AVC decoder, the quantization level *Z* is (normatively) reconstructed to W' by inverse quantization:

$$W' = q \cdot Z \tag{24}$$

[1] It is important to refer that it was decided to always adopt the terminologies used by the authors of the reference papers; this may imply that terms and variables in different sections are not consistent between Sections to be consistent with the corresponding papers.

where *s* is not involved. Therefore, the rounding offset has the key advantage of regulating the quantization process without the need to transmit additional parameters to the decoder.

The selected rate control scheme is the so-called *ρ-domain rate control* due to its superior performance, where *ρ* represents the percentage of zero transform coefficients in each block, although it can be also done at the macroblock level. This algorithm adjusts the *QP* based on the linear rate model:

$$R_c = \Theta(1-\rho) \tag{25}$$

where R_c is the number of coefficient bits, ρ is the percentage of zero DCT coefficients in each block and Θ is the model parameter. To perform ρ-domain rate control in a H.264/AVC encoder, a two pass encoding framework is employed, with the first loop (transform plus quantization) collecting global statistics to determine the final *QP* before the second encoding loop.

Due to their importance for this Thesis, it is important to highlight some of the quantization aspects included in the proposed rate control solution. In [14], an extensive study was made to discover the effect of the rounding offset, *s,* on the bitrate, *R*. If *R* is denoted as $R_c + R_h$, where R_c refers to the transform coefficient bits and R_h refers to the header bits, it can be observed that R_h is almost constant over different *s* and that, with a fixed rounding offset, *s*, only a limited set of R_c values can be obtained with discrete values of *QP*. However, with proper manipulation of *s*, any intermediate number of bits can also be achieved since *s* is a continuous variable. This has been the motivation for the inclusion of the rounding offset in the rate control algorithm.

A linear relationship between $ln(R_c)$ and *s* can be established as:

$$\ln\big(R_c(QP,s)\big) = k_s(QP) \times (s - s_d) + \ln\big(R_c(QP,s_d)\big) \tag{26}$$

where k_s is a model parameter, s_d is the *default rounding offset* and $R_c(QP,s_d)$ is the *resulting bitrate* when encoding with *QP* and s_d. The k_s parameter models how the bitrate changes with *s* when it differs from s_d; it is content specific and its value also depends on the picture type and *QP*. To accommodate its dynamic nature, k_s is estimated for each frame in the proposed ARO algorithm. Only information from previous frames of the same type is used to estimate k_s for a new frame, as it also varies significantly among different picture types. With the estimated k_s based on previous frames and the *QP* derived from a *QP*-based control, the ARO algorithm needs to compute the *s* value that better allows approaching the target bitrate, R_c^T. If the encoding bitrate is $R_c(QP,s_d)$ with a given *QP* and an initial s_d, the rounding offset that better allows approaching R_c^T is computed as:

$$s^T = \frac{1}{k_s} ln \frac{R_c^T}{R_c(QP;s_d)} + s_d \tag{27}$$

Figure 13 presents the block diagram for the proposed ARO rate control algorithm. The algorithm is summarized as follows:

1. ***Initialization and First Frame Coding*** - Set *n* to 1 and initialize k_s and s_d; encode the first frame using s_1^T=s_d at QP_1 that is determined by the ρ-domain rate control algorithm.
2. ***Next Frame Coding*** - Set *n* to *n = n+1* and encode the n^{th} frame with the following steps:
   2.1. The n^{th} frame is pre-processed and a $\rho-(QP, s_d)$ table is build; QP_n is also initialized.
   2.2. $\tilde{R}_n(QP_n, s_d)$ and s_n^T are computed.
   2.3. The n^{th} frame is encoded at QP_n and s_n^T to obtain the encoded bit rate $R_n(QP_n, s_n^T)$.
   2.4. $\widehat{R_n}(QP_n, s_n^T)$ and is computed and $\Theta(QP_n, s_d)$ is updated also the value of k_s is updated using linear regression.
3. A loop to step 2 is applied until all frames are encoded.

For a more detailed description of the main steps of the rate control algorithm and how some of the variables and parameters are obtained, please refer to [14].

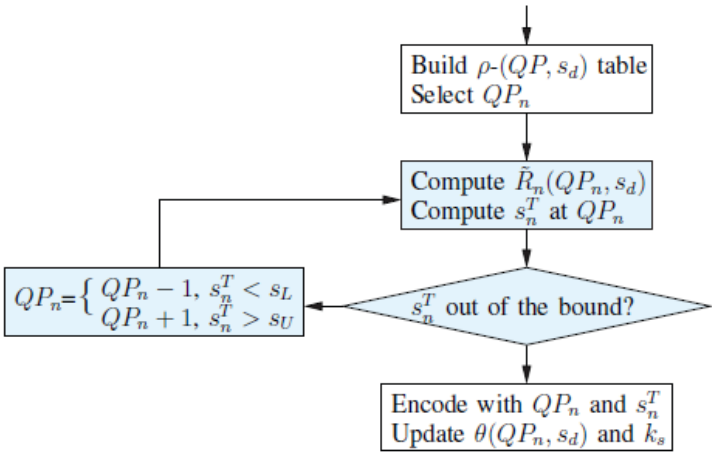


Figure 13 - Block diagram for the ARO algorithm [14].

B. Performance Assessment

To test its performance, the ARO algorithm was implemented in a H.264/AVC encoder. The rounding offset was restricted to be within [0.23,0.45] for Intra frames and [0.05,0.32] for Inter frames. The maximum number of iterations in step 2.2 above was set to $M = 3$. The algorithm was implemented on a frame level to guarantee consistent visual quality throughout all macroblocks. The proposed method has been compared against a QP-based ρ-domain method where a similar rate control is used with the exception of the adaptive rounding offsets in the quantization process. Results for two HD sequences are presented in Figure 14 and in Table 1.

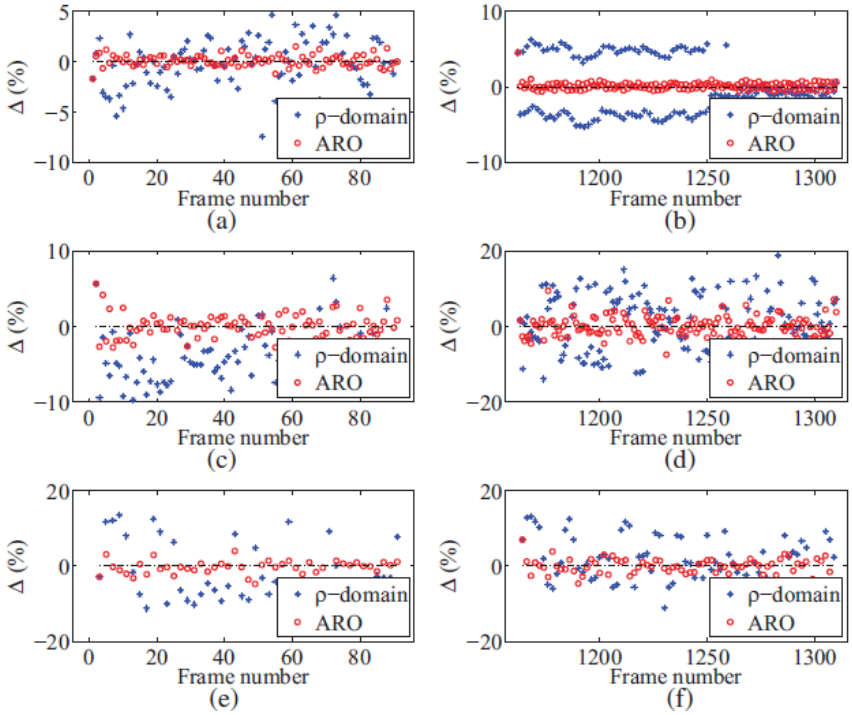


Figure 14 - Relative control errors for the "erin" (left) and "royal" (right) sequences: (a)(b) I in 'III'; (c)(d) P in 'IPP'; (e)(f) B in 'IBP' [14].

Table 1 - $\overline{\Delta}$ and PSNR comparison for HD contents [14].

video	erin			royal4		
	I	P	B	I	P	B
$\overline{\Delta}$ (%)						
ρ-domain	1.97	4.78	5.51	2.98	6.50	5.15
ARO	0.44	1.32	1.50	0.39	1.96	2.32
PSNR (dB)						
ρ-domain	42.82	41.29	40.96	38.73	39.96	39.64
ARO	42.83	41.39	41.03	38.64	39.93	39.62

The results demonstrate that the ARO rate control accuracy is consistently higher than the QP-based ρ-domain method for all tested sequences and conditions. By introducing the adaptive rounding offset method into the rate control process, the control error for most frames is reduced to 0.5% or lower for Intra frames and around 2% for Inter frames. As shown in Table 1, the average control error $\overline{\Delta}$ with ARO algorithm is only 0.4% for I frames, 1.3%-2% for P frames and 1.5%-2.3% for B frames. The $\overline{\Delta}$ parameter is used to measure rate control performance. It is defined for the m^{th} frame as:

$$\Delta_m = \frac{B_m - B_m^T}{B_m^T} \times 100\% \tag{28}$$

where B_m and B_m^T are the actual and target number of bits fo the m^{th} frame. The average control error over N frames is:

$$\overline{\Delta} = \sum_{M=1}^{N} \frac{|\Delta_m|}{N} \tag{29}$$

In summary, the ARO algorithm accomplishes 70% or higher rate control accuracy improvement over the QP-based scheme with almost no additional complexity. For a more detailed description of the tests and results for SD sequences, please refer to [14].

2.4. Reviewing Relevant Background on Correlation Noise Modeling and Optimal Reconstruction

This subsection intends to review the most used solutions for two important problems in distributed video coding: Correlation Noise Modeling and Optimal Reconstruction with multiple side information. These issues are relevant for the development of this Thesis and the solutions here reviewed are those used by the well-known DISCOVER WZ video codec [15].

2.4.1. Correlation Noise Modeling for Efficient Transform Domain Wyner-Ziv Video Coding

In order to make use of the SI obtained in a DVC solution, the decoder needs to have an accurate model to characterize the correlation noise between the original WZ frame and the corresponding SI frame. The correlation noise $WZ - SI$ can be interpreted as a virtual channel with an error pattern characterized by a statistical distribution. This section presents the correlation noise modeling solutions proposed by Brites et al. [16] which are largely used in the DVC literature.

In predictive video coding, the *Laplacian* distribution is used to model the distribution of the motion-compensated residual DCT coefficients [16]. In most DVC solutions, the residual between the original WZ frame and the corresponding SI frame is also modeled by a *Laplacian* distribution such as (where (x, y) is the position to be evaluated within the WZ and SI frames):

$$p[WZ[(x,y) - SI(x,y)] = \frac{\alpha}{2}\exp[-\alpha|WZ(x,y) - SI(x,y)|] \tag{30}$$

where α is the *Laplacian* distribution parameter defined by:

$$\alpha = \sqrt{\frac{2}{\sigma^2}} \tag{31}$$

In this context, a major goal is to design methods to estimate the Correlation Noise Model (CNM) based on the available information. This estimation may be performed i) in the pixel or the transform domain depending if the pixels residuals or DCT transform residuals are modeled and, ii) online or offline, where *online correlation noise modeling* corresponds to a process where the CNM parameters are estimated at the decoder without using original data while, on the contrary, *offline correlation noise modeling* corresponds to a process where the CNM

parameters are obtained at the encoder using the original data and a replica of the side information that is created at the decoder.

The major issue with offline correlation noise estimation is that it is not acceptable from a realistic point of view since it requires the encoder to recreate the SI. Since complex motion estimation and compensation algorithms are used to generate the SI, this task is impossible to perform at the encoder when its complexity is to be kept low. The more realistic approach is to estimate the α parameter at the decoder side, where more computational resources are typically available according to the DVC paradigm; however, this implies that this estimation has to be made without access to the original data. As the most efficient DVC solutions work in the transform domain, only this case will be considered in the following.

A. *Offline Transform-Domain CNM*

As mentioned above, *Transform-Domain CNM* exploits the spatial redundancy within a frame by applying a DCT transform over the frame blocks. Thus, the correlation noise distribution regards now the residual between corresponding DCT bands of the WZ and the corresponding SI frames. Once again, a *Laplacian* distribution is used to model the statistical distribution of the noise distribution. In the same way as for the pixel-domain, there are three granularity modeling levels which are stated in the following; for a more accurate description of each of these solutions, please refer to [16]:

1. ***Correlation Noise Model at DCT Band/Sequence Level:*** This technique models the correlation noise by performing a coarse offline estimation of the *Laplacian* distribution parameter α over the entire sequence, at the DCT band level; this means that α parameter is estimated for each DCT band for the full sequence, leading to 16 different parameters if a 4×4 DCT is used. This is not a very efficient modeling process because it does not exploit the variability of the correlation noise along time and space.
2. ***Correlation Noise Model at DCT Band/Frame Level:*** This approach enables the temporal adaptation of the *Laplacian* distribution along the video sequence as different parameters are estimated to each frame. To offline estimate α parameter of the DCT band *b* at the frame level, α_b, the variance of the DCT band *b* coefficients ${\sigma_b}^2$ has to be first computed. For a detailed description of the process to obtain α_b and ${\sigma_b}^2$, please refer to [16]. Despite this α computation process being more efficient than the one previously described, a better RD performance can still be obtained by exploiting the varying spatial correlation.
3. ***Correlation Noise Modeling at Coefficient /Frame Level:*** This technique has the finest granularity level as the Laplacian distribution is adapted both temporally and spatially. To estimate the α parameter at the coefficient/frame level, σ^2 has to be replaced by another metric because, at this coefficient/frame level, the spatial region of support is only a coefficient and so the residual variance is zero. In this context, and since *T(u,v)* corresponds to the residual between corresponding DCT coefficients of the WZ and SI frames, the square of the DCT coefficient value *T(u,v)* is the measure used to replace σ^2.

B. *Online Transform-Domain CNM*

The Online Transform-Domain CNM case is the most relevant since transform domain DVC solutions are more used as they are more efficient than pixel domain DVC solutions and online solutions are more realistic than offline solutions, as already explained above; for this reason,

the solution in [16] for this case will be presented in more detail. Following the previous structure, two techniques to perform *Online Transform-Domain CNM* are discussed in the following:

1. ***Correlation Noise Estimation at DCT Band/Frame Level:*** This technique performs a temporal adaptation of the α parameter along the video sequence. This means that the α is estimated for each DCT band and its value is updated for each frame, thus varying along the video sequence; for this case, the temporal variation of the correlation noise statistics is taken into account. The five steps of this process are presented in the following:

   - Step 1 generates the residual frame *R* by with the motion compensated versions of the frames X_B and X_F calculated as:

$$R(x,y) = \frac{X_F(x+dx_f, y+dy_f) - X_B(x+dx_b, y+dy_b)}{2} \tag{32}$$

     where $X_B(x+dx_b, y+dy_b)$ and $X_F(x+dx_f, y+dy_f)$ represent the backward and forward motion-compensated frames respectively and *(x,y)* corresponds to the pixel location in the *R* frame.

   - Step 2 computes the variance of the residual frame using:

$$\hat{\sigma}_R{}^2 = E_R[R(x,y)^2] - (E_R[R(x,y)]^2) \tag{33}$$

     where $E_R[\cdot]$ is the expectation operation over the residual frame *R*.

   - Step 3 computes the $|T|$ frame as the absolute value of the corresponding elements in the *T* frame, which is the resulting frame after applying a DCT transform to the *R* frame [13].
   - Step 4 computes the $|T|$ frame DCT band *b* variance $\sigma_b{}^2$ as:

$$\sigma_b{}^2 = E_b\left[|T|_b{}^2\right] - (E_b[|T|_b])^2 \tag{34}$$

   - In Step 5 the DCT band *b* $\hat{\alpha}_b$ parameter is estimated as:

$$\hat{\alpha}_b = \sqrt{\frac{2}{\hat{\sigma}_b{}^2}} \tag{35}$$

2. ***Correlation Noise Estimation at Coefficient/Frame Level:*** In this technique, the *Laplacian* distribution parameter is adapted both temporally and spatially, i.e., for each DCT coefficient inside the DCT coefficients frame, *T*. The DCT coefficients are classified as *inlier* or *outlier coefficients.* The inlier coefficients are those coefficients whose value is close to the corresponding DCT band average value, $\hat{\mu}_b{}^2$, and the outlier coefficients are those coefficients whose value is far away from $\hat{\mu}_b$. To determine the degree of proximity between a certain coefficient and the corresponding $\hat{\mu}_b{}^2$, the distance between the coefficient and $\hat{\mu}_b$

is compared with the DCT band variance; this is a good approach since the variance is a measure of how the DCT coefficient values are spread regarding the average value. The first four steps of this method are similar to the corresponding ones for CNM at DCT band/frame level and thus are not repeated here. In step 5, the $|T|$ frame (u, v) DCT coefficient distance D_b is computed using:

$$D_b(u,v) = |T|_b(u,v) - \hat{\mu}_b \tag{36}$$

where $|T|_b(u,v)$ represents the DCT coefficient at the (u, v) position of the $|T|$ frame DCT band *b*. In step 6, the α parameter for the DCT coefficient located at (u, v) position is estimated using:

$$\hat{\alpha}_b(u,v) = \begin{cases} \hat{\alpha}_b, & [D_b(u,v)]^2 \leq {\hat{\sigma}_b}^2 \\ \sqrt{\dfrac{2}{[D_b(u,v)]^2}}, & [D_b(u,v)]^2 > {\hat{\sigma}_b}^2 \end{cases} \tag{37}$$

In (37) two situations can occur: 1) the distance $[D_b(u,v)]^2$ is less than or equal to ${\hat{\sigma}_b}^2$, which corresponds to a region well interpolated; 2) the distance $[D_b(u,v)]^2$ is greater than ${\hat{\sigma}_b}^2$, which corresponds to a block where the residual error is high, meaning that the SI generation process failed for this block.

CNM Performance Comparison

Regarding *Offline Transform Domain* CNM, four QCIF video sequences were selected to test the RD performance at 30 Hz and 15 Hz. The key frames were Intra Coded with H.264/AVC and a GOP size of 2 was used. The various RD points are defined by a 4×4 WZ quantization matrix. The results in [16] show that pursuing a coarser to finer strategy in the α parameter computation leads to consistent RD performance improvements. For all the test sequences used, the WZ RD performance was always above the H.264/AVC Intra curve, independently of the granularity level used to calculate the α parameter.

For *Online Transform Domain* CNM, the fifth and sixth rows in Table 2 show the bitrate saving when the finer granularity level is used instead of a coarser approach. The minus sign in the *Online ΔRate* and the *Offline ΔRate* rows means that, using the coefficient/frame level, there is a rate saving compared with the case where DCT band/frame level is used. The RD points 1-8 correspond to the eight quantization matrices mentioned above. The results presented in Table 2 show that there is a bitrate decrease as the online estimation granularity gets finer. Rate savings between 0.47 kbps and 12.69 kbps, for the first and eight points, respectively, are achieved. As expected, the online correlation noise estimation algorithms have a small coding efficiency loss when compared to the offline models.

Table 2 - DCT Band-Level and Coefficient-Level RD performance for Flower Garden, Foreman, Coastguard and Hall Monitor QCIF Sequences [16].

QCIF Sequences @ 30Hz	Flower Garden								Foreman							
	RD 1 (PSNR = 26.26 dB)	RD 2 (PSNR = 26.98 dB)	RD 3 (PSNR = 27.69 dB)	RD 4 (PSNR = 30.05 dB)	RD 5 (PSNR = 30.87 dB)	RD 6 (PSNR = 31.80 dB)	RD 7 (PSNR = 33.53 dB)	RD 8 (PSNR = 36.28 dB)	RD 1 (PSNR = 30.58 dB)	RD 2 (PSNR = 31.13 dB)	RD 3 (PSNR = 31.71 dB)	RD 4 (PSNR = 33.23 dB)	RD 5 (PSNR = 33.76 dB)	RD 6 (PSNR = 34.62 dB)	RD 7 (PSNR = 36.57 dB)	RD 8 (PSNR = 39.53 dB)
	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]
DCT band (online)	219.31	253.89	288.92	425.42	476.30	561.97	687.98	965.19	162.68	186.56	210.46	288.15	318.74	391.71	511.00	787.21
Coefficient (online)	219.59	253.50	287.51	421.71	472.72	557.03	681.02	958.18	162.54	185.98	209.80	286.20	316.13	387.14	504.79	774.52
DCT band (offline)	219.90	254.23	289.26	425.48	476.38	561.30	686.78	959.82	163.09	186.68	210.64	288.59	319.35	392.67	512.07	787.30
Coefficient (offline)	215.52	248.30	281.31	408.88	457.57	533.57	652.77	908.93	158.77	181.51	204.20	275.32	303.14	367.19	482.68	741.32
Online ΔRate (coeff. – band)	0.28	-0.39	-1.41	-3.71	-3.58	-4.94	-6.96	-7.01	-0.14	-0.58	-0.66	-1.95	-2.61	-4.57	-6.21	-12.69
Offline ΔRate (coeff. – band)	-4.38	-5.93	-7.95	-16.60	-18.81	-27.73	-34.01	-50.89	-4.32	-5.17	-6.44	-13.27	-16.21	-25.48	-29.39	-45.98
Band ΔRate [%] (on. – off.)/off.	-0.27	-0.13	-0.12	-0.01	-0.02	0.12	0.17	0.56	-0.25	-0.06	-0.09	-0.15	-0.19	-0.24	-0.21	-0.01
Coeff. ΔRate [%] (on. – off.)/off.	1.89	2.09	2.20	3.14	3.31	4.40	4.33	5.42	2.37	2.46	2.74	3.95	4.29	5.43	4.58	4.48
QCIF Sequences @ 15Hz	Coastguard								Hall Monitor							
	RD 1 (PSNR = 28.45 dB)	RD 2 (PSNR = 29.13 dB)	RD 3 (PSNR = 29.57 dB)	RD 4 (PSNR = 30.30 dB)	RD 5 (PSNR = 30.90 dB)	RD 6 (PSNR = 31.63 dB)	RD 7 (PSNR = 32.98 dB)	RD 8 (PSNR = 35.28 dB)	RD 1 (PSNR = 31.37 dB)	RD 2 (PSNR = 31.86 dB)	RD 3 (PSNR = 32.56 dB)	RD 4 (PSNR = 34.13 dB)	RD 5 (PSNR = 34.78 dB)	RD 6 (PSNR = 35.66 dB)	RD 7 (PSNR = 37.00 dB)	RD 8 (PSNR = 39.82 dB)
	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]	Rate [kbps]
DCT band (online)	80.19	98.16	109.65	142.11	159.91	199.24	260.72	412.55	84.30	94.16	105.04	134.17	145.54	172.02	202.79	296.55
Coefficient (online)	80.33	98.13	109.29	140.68	158.01	196.45	255.91	403.94	83.83	93.87	104.26	132.45	143.42	169.57	198.08	287.02
DCT band (offline)	80.05	97.88	109.19	141.65	159.65	198.68	259.30	409.38	84.25	94.03	105.16	134.12	145.29	171.87	202.19	295.70
Coefficient (offline)	78.48	95.66	106.74	134.51	151.69	184.90	241.77	381.02	83.04	92.48	103.13	129.45	140.92	165.06	191.79	277.02
Online ΔRate (coeff. – band)	0.14	-0.03	-0.36	-1.43	-1.90	-2.79	-4.81	-8.61	-0.47	-0.29	-0.78	-1.72	-2.12	-2.45	-4.71	-9.53
Offline ΔRate (coeff. – band)	-1.57	-2.22	-2.45	-7.14	-7.96	-13.78	-17.53	-28.36	-1.21	-1.55	-2.03	-4.67	-4.37	-6.81	-10.40	-18.68
Band ΔRate [%] (on. – off.)/off.	0.17	0.29	0.42	0.32	0.16	0.28	0.55	0.77	0.06	0.14	-0.11	0.04	0.17	0.09	0.30	0.29
Coeff. ΔRate [%] (on. – off.)/off.	2.36	2.58	2.39	4.59	4.17	6.25	5.85	6.02	0.95	1.50	1.10	2.32	1.77	2.73	3.28	3.61

2.4.2. Optimal Reconstruction in WZ Video Coding with Multiple Side Information

This subsection presents an optimal reconstruction approach, largely used in the DVC literature and also in the DISCOVER WZ video codec, proposed by Kubasov et al. [17], which exploits the actual correlation model between the source and the side information; to minimize the mean squared error (MSE) of the reconstructed samples after the Slepian-Wolf decoder provides the decoded quantization bin. Given the *Laplacian* correlation model, a closed form expression of the reconstructed value is derived. This process allows gains up to 1 dB in RD performance with minimum costs in terms of decoder complexity, when compared with a straightforward reconstruction approach as defined below. The Wyner-Ziv coding solution used here is identical to the one described in Section 2.2 of this report, but for more detail on that subject please refer to [17].

A. Optimal Reconstruction Method with a Single SI

Regarding the *Minimum MSE Reconstruction* process to be presented here, it is important to provide some notation first. Let *M* denote the number of quantized levels and $z_0 < z_1 < \cdots < z_M$ denote the quantizer levels themselves. Since the quantizer is uniform, $z_{i+1} - z_i = \Delta, \forall_i = 0, \ldots, M-1$, where Δ is the quantization step size. A straightforward approach to reconstruct the source coefficient *x* using a side information value *y* may be:

$$\hat{x}_{opt} = \begin{cases} z_i, & y < z_i \\ y, & y \in [z_i, z_{i+1}] \\ z_{i+1}, & y > z_{i+1} \end{cases} \tag{38}$$

where $\hat{x}$ denotes the reconstructed value and *i* the quantization index of *x*. Although there are other reconstruction approaches in the literature [17], they are all suboptimal as the one in (38).

The optimal reconstruction approach is to compute $\hat{x}$ as the expectation $E[x|x\ \varepsilon[z_i, z_{i+1}], y]$ of the random variable $\hat{x}$ given the quantization interval $[z_i, z_{i+1}]$ and the side information value *y*:

$$\hat{x}_{opt} = E[x|x\ \varepsilon[z_i, z_{i+1}], y] = \frac{\int_{z_i}^{z_{i+1}} x f_{x|y}(x)dx}{\int_{z_i}^{z_{i+1}} f_{x|y}(x)dx} \tag{39}$$

where $f_{x|y}(x)$ is the conditional p.d.f of *x* given *y* which corresponds to the correlation noise model that characterizes the relationship between x and y This reconstructed value $\hat{x}_{opt}$ corresponds to the minimum mean-squared error estimate of the source *x* [17]. To avoid numerical computation of integrals, the *Laplacian* model of the residue between the source DCT band *x* and the side information DCT band *y* is used and the following derived closed form is used:

$$\hat{x}_{opt} = \begin{cases} z_i + \dfrac{1}{\alpha} + \dfrac{\Delta}{1 - e^{\alpha\Delta}}, & y < z_i \\ y + \dfrac{\left(\gamma + \dfrac{1}{\alpha}\right) e^{-\alpha\gamma} - (\delta + \dfrac{1}{\alpha}) e^{\alpha\delta}}{2 - (e^{-\alpha\gamma} - e^{\alpha\delta})}, & y\ \varepsilon\ [z_i, z_{i+1}] \\ z_{i+1} - \dfrac{1}{\alpha} - \dfrac{\Delta}{1 - e^{\alpha\Delta}}, & y \geq z_{i+1} \end{cases} \tag{40}$$

where $\gamma = y - z_i$ and $\delta = z_{i+1} - y$. Comparing (38) with (40), it is possible to observe that the reconstruction levels are shifted towards the center of the quantization interval. When $\alpha = 0$, this means when *y* conveys no information about *x*, $\hat{x}_{opt} = z_i + z_{i+1}/2$; on the other hand, when $\alpha \to \infty$, $\hat{x}_{opt}$ approaches $\hat{x}$ in (38).

B. Optimal Reconstruction Method with Multiple SI

Regarding a scenario with *Multiple Side Information,* a case referring to two side information hypothesis obtained with different motion-compensated temporal interpolation (MCTI) methods, named the block-based MCTI and the mesh-based MCTI, is presented in [17]. Figure 15 presents the structure of the decoder with multiple side information, where Y_1 and Y_2 are SI both considered correlated with the source *X*.

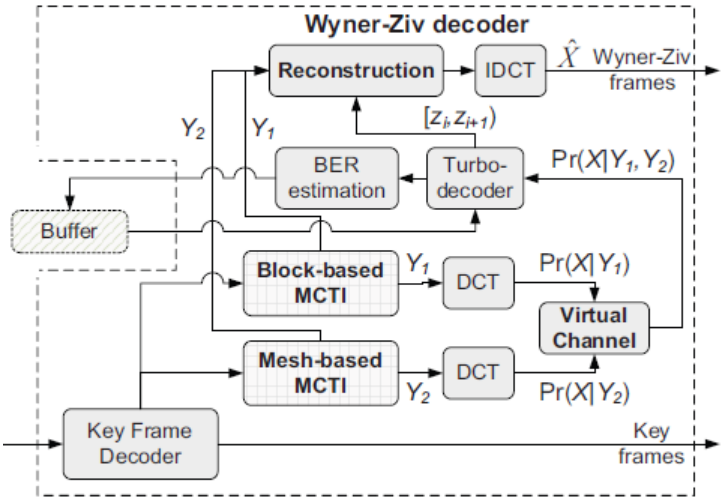


Figure 15 - Decoder Structure with Multiple Side Information [17].

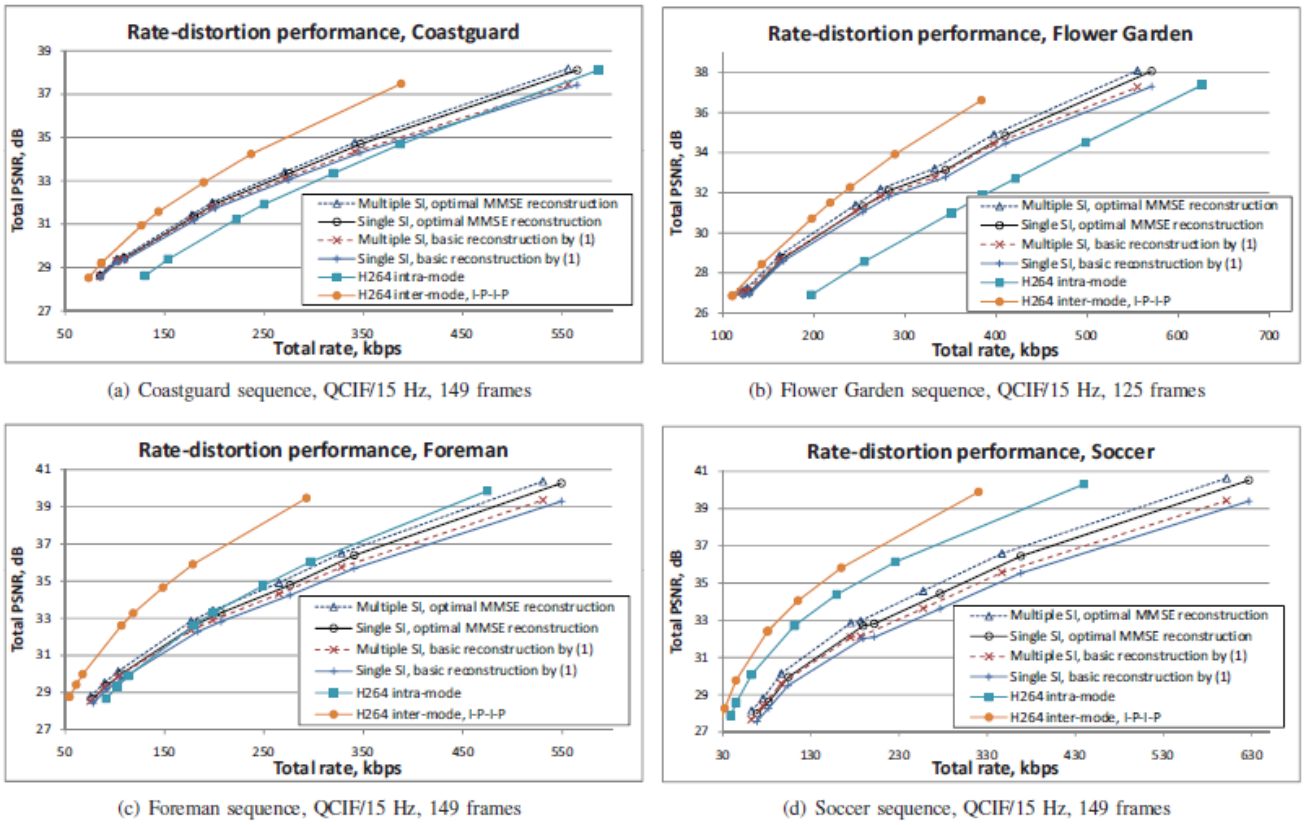
The reconstruction module requires some adaptation for the multiple side information case. Here the optimal minimum MSE is given by (39) using $f_{X|Y_1,Y_2}(x)$ instead of $f_{X|Y}(x)$. It is here assumed that $f_{X|Y_1,Y_2} = \frac{1}{2}[f_{x|Y_1}(x) + f_{X|Y_2}(x)]$, meaning that the two SIs have the same weight but this could be done differently. The reconstructed value with multiple side information is then:

$$\hat{x}_{opt}, MH = \frac{\sum_{k=1}^{2}\sum_{j=0}^{s-1}\left[\frac{\alpha_k}{2}\int_{q_j}^{q_{j+1}} x e^{-\alpha k|x-y_k|}\right]dx}{\sum_{k=1}^{2}\sum_{j=0}^{s-1}\left[\frac{\alpha_k}{2}\int_{q_j}^{q_{j+1}} e^{-\alpha k|x-y_k|}\right]dx} \quad (41)$$

where (41) represents a simplified expression available in [14].

C. Performance Assessment

The RD performance of the optimal reconstruction method described above was assessed with several sequences at QCIF resolution, 15 Hz with a *Group of Frames* (GOF) size 2. Figure 16 shows the average PSNR of the luminance component for both the key and WZ frames versus the total bitrate. The method is compared with the straightforward reconstruction approach presented in (38).



(a) Coastguard sequence, QCIF/15 Hz, 149 frames

(b) Flower Garden sequence, QCIF/15 Hz, 125 frames

(c) Foreman sequence, QCIF/15 Hz, 149 frames

(d) Soccer sequence, QCIF/15 Hz, 149 frames

Figure 16 - RD performance with optimal MMSE reconstruction [17].

The results show that for both the single and multiple side information scenarios the proposed method gains up to 1 dB in PSNR when compared with the straightforward reconstruction method; moreover, the usage of multiple side information improves the RD performance by up to 0.3 dB regarding the single SI case. Using the optimal MMSE reconstruction method, increases the decoding time with single side information only by 0.01 seconds per frame, when compared with a simple reconstruction method [17].

Chapter 3

Predictive Video Coding with Statistical Reconstruction: Video Codec Architecture

This chapter intends to present the architecture of the video coding solution proposed in this Thesis, highlighting the novel modules inserted in the *standard* H.264/AVC codec. While, formally speaking, there are only standard H.264/AVC decoders as encoders are not normative, there is also a basic H.264/AVC encoder architecture which is typically used. Naturally, especial emphasis will be given to the new modules added to the *standard* H.264/AVC architecture which will be presented by detailing their main functionalities. As stated in Chapter 1, the objective of the proposed video coding solution is to exploit the correlation noise statistics between the source and the prediction taken as the side information to improve the DCT coefficients reconstruction values and thus the final RD performance, notably by minimizing the error of the reconstructed samples.

The basic idea of this improved predictive video codec is to take a typical distributed video coding tool, in this case the statistical reconstruction at the decoder, and use it to obtain more faithfully reconstructed, this means dequantized, DCT coefficients. In this context, first the pixel based prediction based on the received prediction modes and motion vectors (which is playing the role of the side information in a distributed video codec) has to be brought to the DCT domain which is made using a QP=0 quantization step to avoid introducing any quantization artifacts since the best 'side information' is desired; the quantization process is applied as in the H.264/AVC codec, which means that the transform and quantization cannot be separated. After, the decoder has to statistically characterize the prediction residual which is received at the decoder; this is very different from a distributed codec where no residual is ever received and thus it has to be estimated at the decoder. Finally, the final decoded frame is obtained by using a MMSE reconstruction function typical to the DVC codecs. It is important to stress that, in a first step, only a quantization bin is defined for the decoded residual DCT coefficients and the precise reconstructed values depend on the reconstruction strategy. Two quantization strategies for H.264/AVC have already been presented, notably the rigid standard H.264/AVC

quantization strategy in Sections 2.3.1 and 2.3.2 and the flexible ARO quantization in Section 2.3.3. Both these strategies guarantee H.264/AVC compliance as the reconstruction function at the decoder is the same normative function. This does not happen for the solution proposed in this Thesis where a decoder tools is changed and thus H.264/AVC compliance lost.

The high-level encoder and decoder architectures of the proposed predictive video codec with statistical reconstruction are presented in Figure 17 and Figure 18. The architecture is based on a *standard* H.264/AVC codec design with the addition of three novel modules labeled as *Optimal Transform, Scaling and Quantization, Residual Statistical Modeling* and *Statistical Reconstruction*. As this is a predictive codec where the encoder and decoder have to be always in perfect prediction synchronism, all modules inserted at the decoder have to be replicated at the encoder as all predictive encoders include the corresponding predictive decoder to guarantee the mentioned synchronism.

As the H.264/AVC video codec walkthrough has already been presented in Section 2.1.2, here an overview of the flow of the coding process is presented with emphasis on the three additional modules that are briefly described in the following:

- **Picture Partitioning into 16×16 macroblocks –** Partition of the picture into fixed-sized macroblocks.
- **Intra and Inter - Frame Prediction –** Creation of the prediction modes in the spatial and temporal domains.
- **Transform Scaling and Quantization –** Exploitation of the spatial and temporal redundancy by transforming and quantizing the prediction residual.
- **Optimal Transform, Scaling and Quantization –** This module which is common to both the encoder and decoder takes the pixel based prediction created with the available/received coding modes and motion vectors and brings it to the DCT domain while quantizing it with a QP equal to zero to avoid any associated quantization error, this means obtaining the best 'side information'. This DCT domain prediction is necessary to determine the bin where the final DCT coefficient should be reconstructed, and it is quantized also to make sure that there are no mismatches in scaling factors between the prediction and the residual.
- **Residual Statistical Modeling –** The objective of this module is to statistically characterize the correlation noise this means the (residual) DCT coefficients; here, the correlation noise corresponds to the quantized and DCT transformed residual between the original and the predicted frames; as this residual is transmitted to the decoder, it is possible to statistically analyze and characterize it in a very precise way. The residual statistical modeling process considers two steps:
    1. **DCT Coefficients Statistical Analysis** This first section aims simply at confirming that a Laplacian distribution may in fact accurately characterize the prediction residual. With this purpose histograms representing the prediction residual were plotted alongside the best Laplacian curve, i.e. the one minimizing the MMSE.
    2. **Statistical Model Parameter Computation** – Following the usual approach in the literature, the residual is after fitted to a Laplacian distribution which is used to model the distribution of the residual DCT coefficients; The objective is to obtain the so-called *alpha Laplacian parameter*, α, i.e. for each DCT band coefficient, the correlation noise is modeled in a process similar to the process described in Section 2.4.1 for distributed video coding. The obtained alpha parameters are used in the Statistical Reconstruction

module both at the decoder and encoder to obtain the reconstructed residuals while avoiding any mismatch or drift.

- **Statistical Reconstruction** – This module has the target to reconstruct the quantized DCT coefficients in a different way from the NURQ reconstruction solution with p=0 which is typically used in H.264/AVC (see equation (4) in Section 2.3.1). This module employs a statistical reconstruction method typically used in DVC codecs to substitute the usual inverse quantization in the H.264/AVC decoder with the expectation of improving the RD performance; as mentioned before, formally speaking, this change makes the proposed decoder non H.264/AVC compliant.
- **In – Loop Deblocking Filter –** Used to eliminate visible block structures, the so called *block-effect*.
- **Entropy Coding –** CABAC is selected to provide additional efficiency.

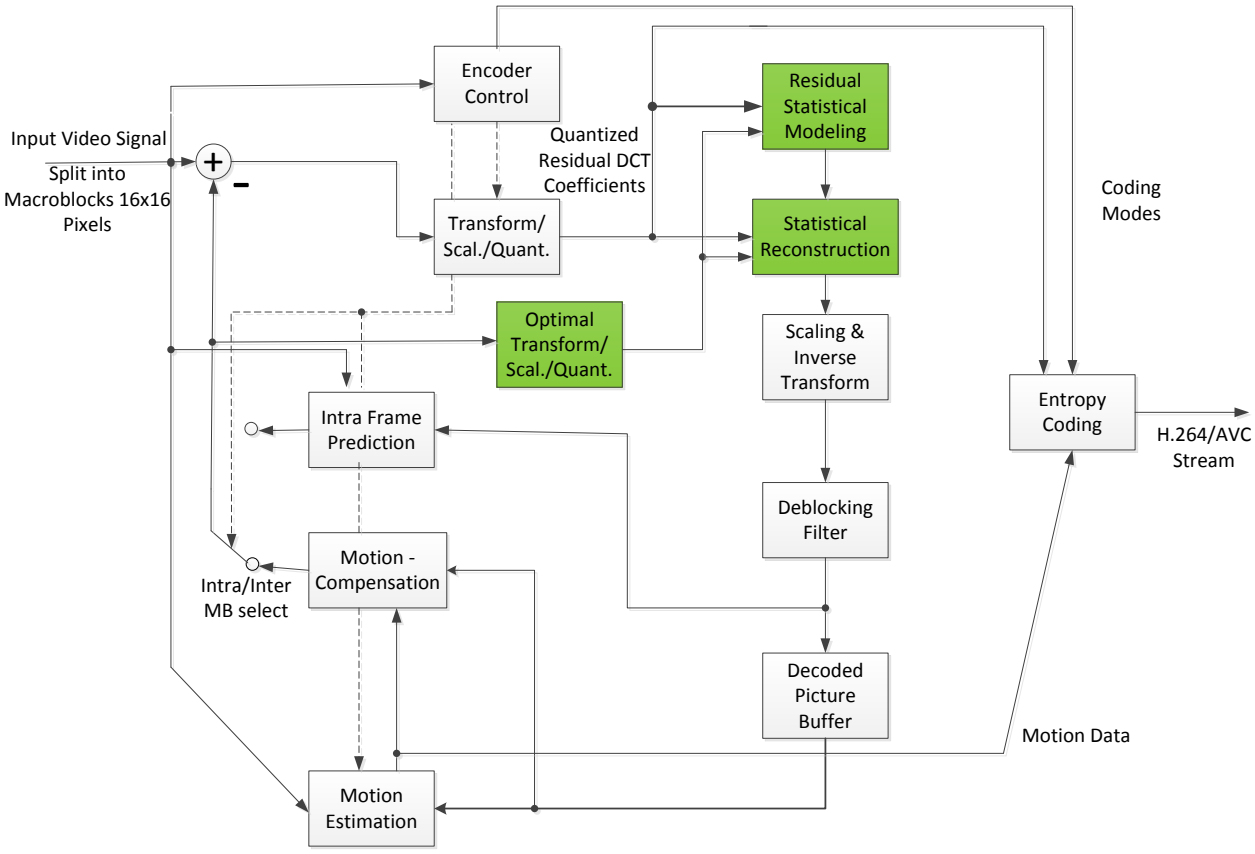


Figure 17 - High-level encoder architecture of the proposed video coding solution.

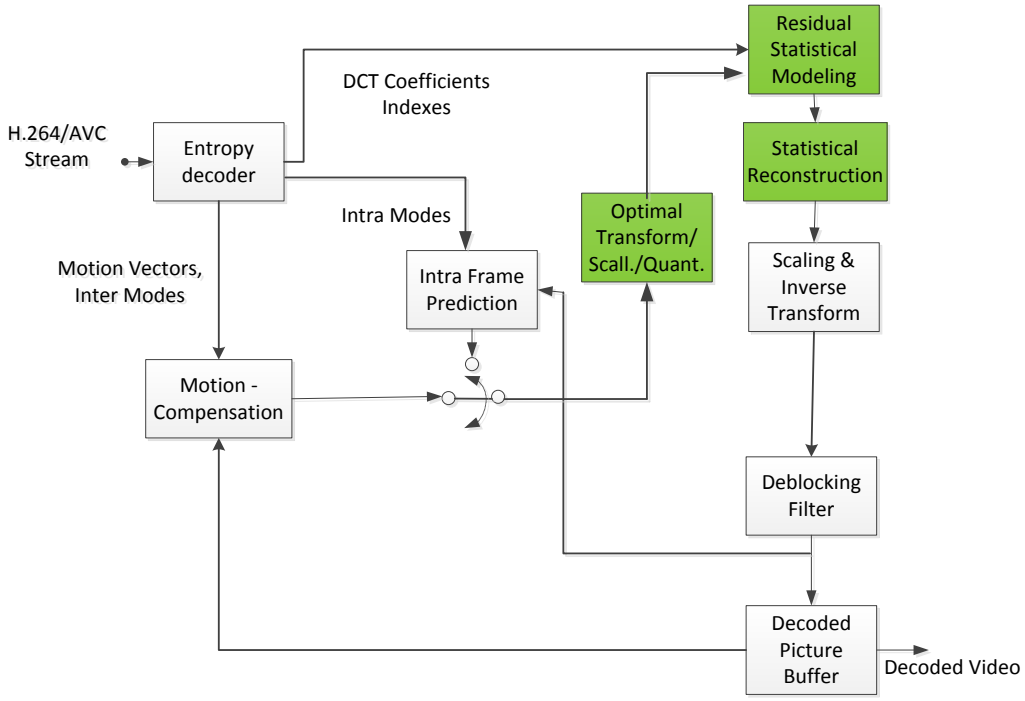


Figure 18 – High-level decoder architecture of the proposed video coding solution.

The proposed video coding solution aims to provide a different reconstruction approach for H.264/AVC video decoder as opposed to using the typical inverse quantization defined by the H.264/AVC standard. After presenting in this chapter an overview of the architecture designed in this Thesis, Chapter 4 will present the various novel modules in depth to provide a better understanding of the overall solution.

Chapter 4

Predictive Video Coding with Statistical Reconstruction: Novel Coding Tools

This chapter provides a detailed description of the novel tools developed in the context of the video coding solution proposed in this Thesis. This Chapter is divided in three sections, each of them presenting the operational details associated to the three novel modules: *Optimal Transform, Scaling and Quantization* module (Section 4.1), *Residual Statistical Modeling* module (Section 4.2) and *Statistical Reconstruction* module (Section 4.3). These modules, which are present at both the encoder and decoder sides since a predictive coded is used, are described according to their order of appearance in the video codec walkthrough presented in Chapter 3.

4.1. Optimal Transform, Scaling and Quantization

As explained in Chapter 3, the *Optimal Transform, Scaling and Quantization* module aims at converting the pixel based prediction (created with the encoder available and decoder received coding modes and motion vectors) to the DCT domain while quantizing it with QP = 0, i.e. the finest quantization possible, thus the attribute 'optimal'. This DCT domain prediction avoids any associated quantization error, thus allowing to obtain the best transform domain 'side information', while reusing the transform, scaling and quantization operations performed in a regular H.264/AVC codec. Since the proposed statistical reconstruction operation is performed in the DCT domain (see Section 4.3), the DCT domain prediction quantization also avoids mismatches in the scaling factors between the prediction and the residual data, both involved in the proposed statistical reconstruction solution.

In this context, each 4×4 luma block within a prediction macroblock is first transformed using a 4×4 integer DCT transform. After, the prediction DCT coefficients, P, are scaled and quantized according to (42) and (43), generating a 4×4 block of quantized prediction DCT coefficients, P_q; this procedure is similar to the one applied to the residual DCT coefficients in the regular quantization process used in the H.264/AVC reference software [13] (see Chapter 2).

$$\left|P_q(u,v)\right| = \left(|P(u,v)| \times A(Q_M,u,v) + f \times 2^{15+Q_E}\right) \gg (15+Q_E), \quad Q_M = \text{QP} \bmod 6, \quad Q_E = QP/6 \tag{42}$$

$$\text{sign}\{P_q(u,v)\} = \text{sign}\{P(u,v)\} \tag{43}$$

In (42), (u,v) stands for the DCT coefficient position within the 4×4 block, $A(Q_M,u,v)$ is a H.264/AVC tabled value associated to the quantization operation, which depends on the QP value and DCT coefficient position, and f is the parameter controlling the quantization bin width around zero (the so-called *dead-zone*); typically, f is 1/3 for Intra blocks and 1/6 for Inter blocks [13]. In (42), $\gg$ represents a binary shift right, which is equivalent to a division operation in integer arithmetic, and $\bmod$ stands for the modulus operator, which returns the division remainder. In (43), sign{x} stands for the signal operator, which returns 1 (resp. -1) when $x > 0$ (resp. $x < 0$) and 0 when $x = 0$.

Since the prediction DCT coefficients are quantized with QP = 0, Q_M=0 and Q_E = 0, and thus any associated quantization error is avoided, allowing to obtain the best transform domain 'side information'. The quantized prediction DCT coefficient $P_q(u,v)$ is also known as *level* or *quantization bin*. The inverse quantization (or reconstruction) of the prediction DCT coefficients is obtained from:

$$P_r(u,v) = \left\{\left[P_q(u,v) \times B(Q_M,u,v)\right] \ll Q_E + 2^3\right\} \gg 4 \tag{44}$$

where $B(Q_M,u,v)$ is a H.264/AVC tabled value associated to the inverse quantization operation, which depends on the QP value and DCT coefficient position. The inverse quantized prediction DCT coefficients P_r will be later used by the Statistical Reconstruction module to obtain the reconstructed DCT coefficients (see Section 4.3).

4.2. Residual Statistical Modeling

As mentioned in Chapter 3, the main objective of this module is to statistically characterize the correlation noise, i.e. the residual DCT coefficients. In the context of the proposed video coding solution, the correlation noise corresponds to the quantized and DCT transformed residual/difference, R_q, between the original and the predicted frames. Since the quantized residual is transmitted to the decoder, it is possible to analyze and characterize it both at the encoder and decoder (which does not happen in distributed video coding). Thus, this module makes use of R_q to obtain the statistical parameter characterizing the statistical function adopted to model the residual for each DCT coefficient.

4.2.1. DCT Coefficients Statistical Analysis

The Laplacian distribution (see (45)) is typically used to model the distribution of the residual DCT coefficients in predictive video coding [18]. Although more accurate models can be found in the literature, such as the generalized Gaussian distribution [19], the Laplacian distribution constitutes a good tradeoff between model accuracy and complexity and, therefore, it is often chosen:

$$p(R) = \frac{\alpha}{2} \times \exp(-\alpha \times |R|) \tag{45}$$

In (45), $p(.)$ stands for the probability density function, R for the residual DCT coefficient and α for the Laplacian distribution parameter. This first section aims at confirming the

appropriateness of this type of distribution for the residual DCT coefficients and thus does not correspond to an operational module in the codec.

Figure 19 to Figure 24 depict the actual histogram of the residual R for the I, P and B frames, respectively, for the sequences City and Night (151 frames) at 1280×720 spatial resolution, 60 Hz, GOP 15 with IBBPBBP prediction structure. The Laplacian distribution resulting from curve fitting to the histogram has determined α values equal to $3{,}75 \times 10^{-3}$, $1{,}21 \times 10^{-2}$ and $1{,}43 \times 10^{-2}$ for the I, P and B frames, respectively, for the *City* sequence, and $5{,}9 \times 10^{-3}$, $1{,}36 \times 10^{-2}$ and 3×10^{-2} for the I, P and B frames, respectively, for the *Night* sequence; in Figure 19 to Figure 24, both the histogram and the Laplacian using the mentioned parameters fitting are shown.

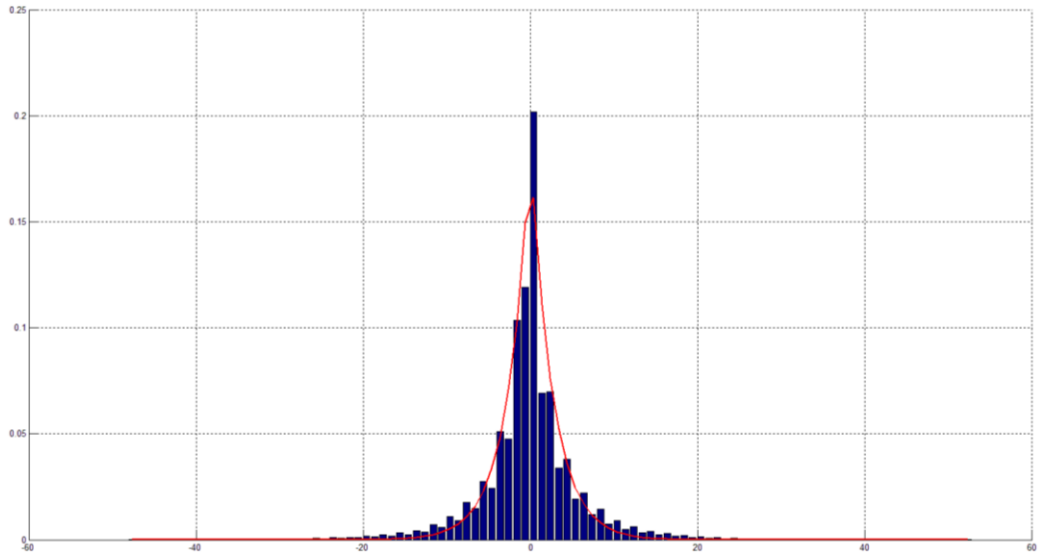


Figure 19 - Residual histogram and Laplacian fitting for the I frames AC1 band of the sequence City, 1280x720, 60 Hz, QP = 10.

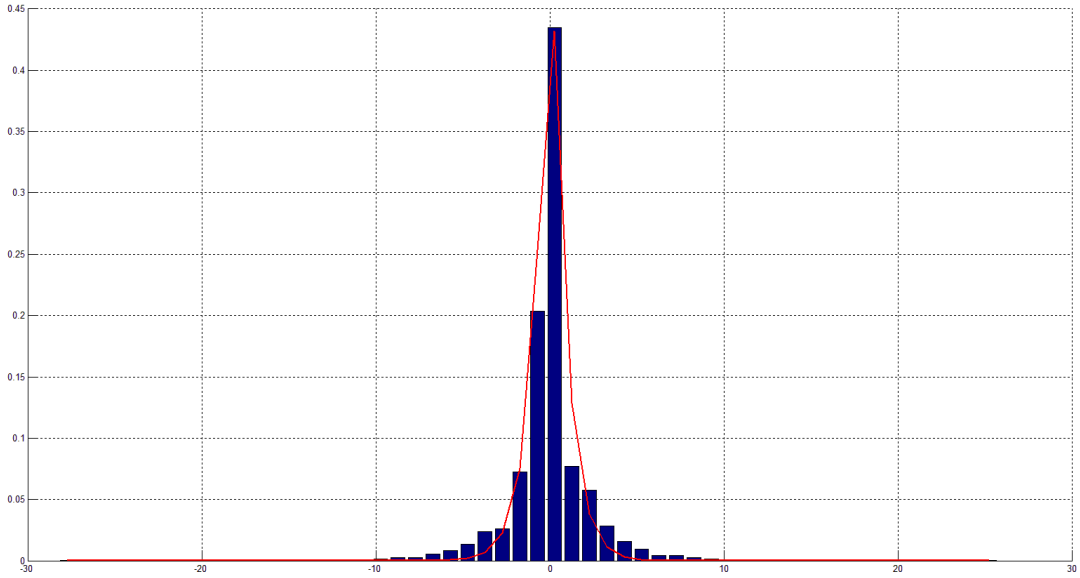


Figure 20 - Residual histogram and Laplacian fitting for the P frames AC5 band of the sequence City, 1280x720, 60 Hz, QP = 11.

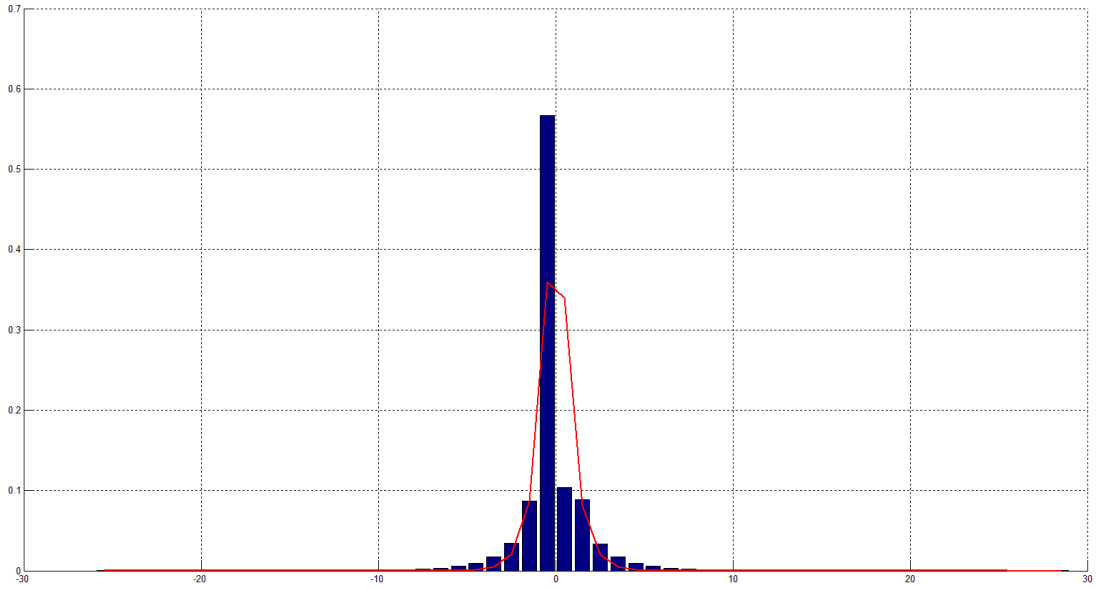


Figure 21 - Residual histogram and Laplacian fitting for the B frames AC9 band of the sequence City, 1280x720, 60 Hz, QP = 12.

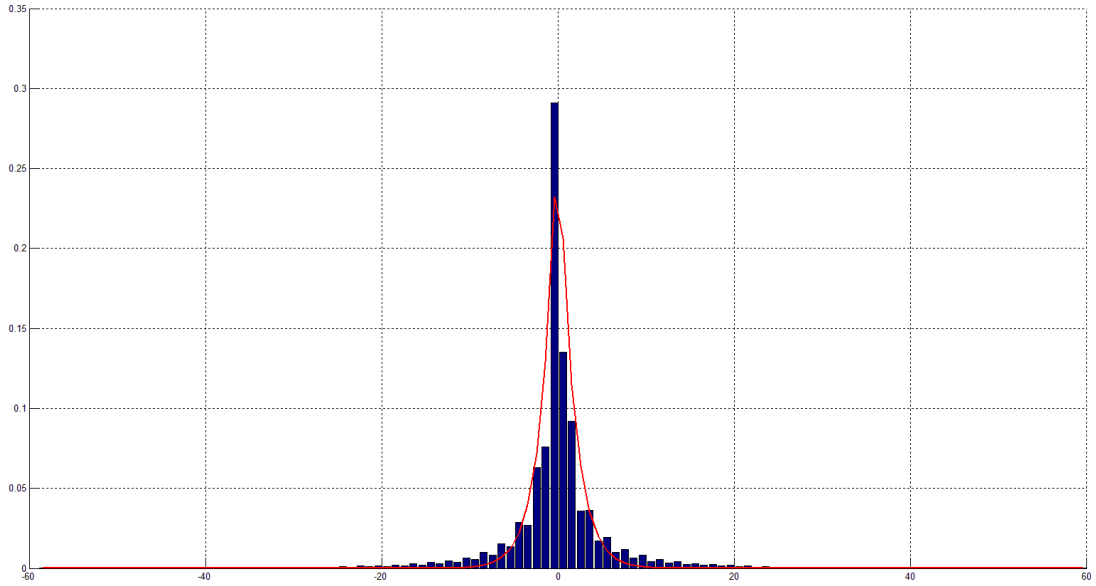


Figure 22 - Residual histogram and Laplacian fitting for the I frames AC1 band of the sequence Night, 1280x720, 60 Hz, QP = 10.

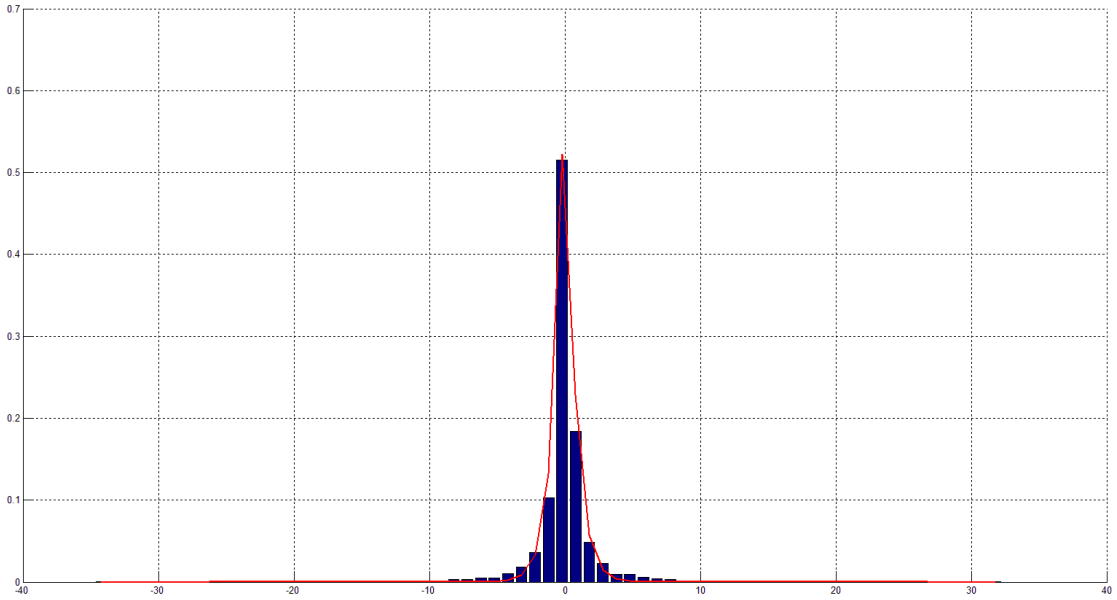


Figure 23 - Residual histogram and Laplacian fitting for the P frames AC5 band of the sequence Night, 1280x720, 60 Hz, QP = 11.

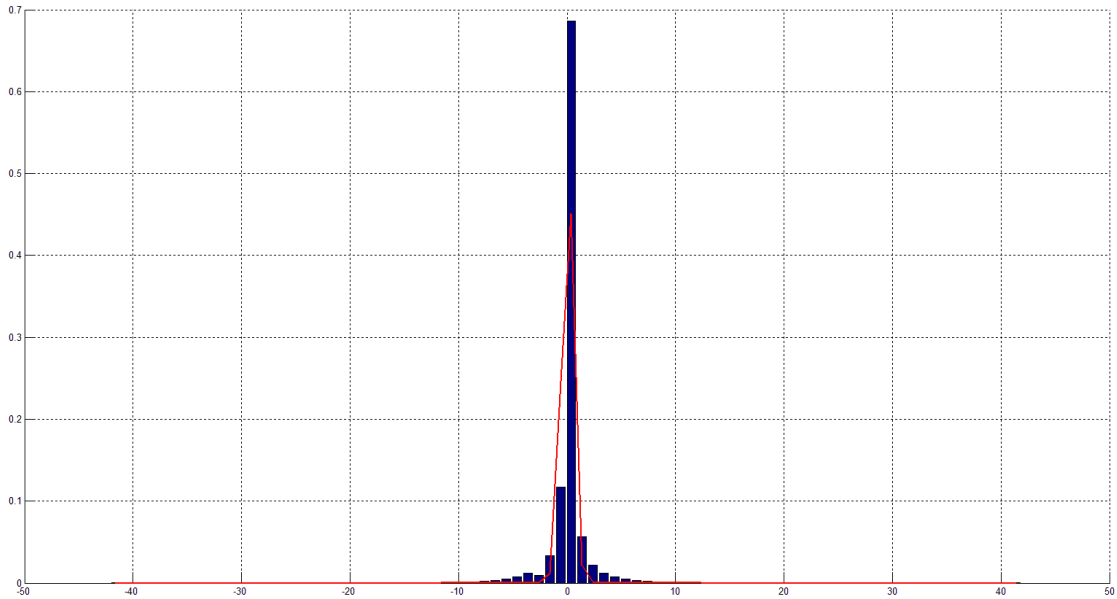


Figure 24 - Residual histogram and Laplacian fitting for the B frames AC9 band of the sequence Night, 1280x720, 60 Hz, QP = 12.

The hypothesis that the Laplacian distribution fits the residual histogram has been validated by using the Chi-square goodness-of-fit test [20] which is a popular test to measure how well a specific statistical distribution fits a set of observed measures. Chi-square values of $7{,}1 \times 10^{-3}$, 3×10^{-3} and $8{,}6 \times 10^{-2}$ were obtained for the I, P and B frames, respectively, of the *City* sequence, corresponding to Figure 19 to Figure 21. For the *Night* sequence, Chi-square values of $1{,}6 \times 10^{-2}$, $1{,}6 \times 10^{-3}$ and $8{,}2 \times 10^{-2}$ were obtained for the I, P and B frames, respectively, corresponding to Figure 22 to Figure 24. For a typical significance level of 5% [20], those values imply accepting the hypothesis that the residual DCT coefficients follow a Laplacian distribution; the significance level is the probability of rejecting an hypothesis although it is true [20]. The same conclusion has been obtained for more video sequences and test conditions evaluated in this Thesis.

4.2.2. Statistical Model Parameter Computation

After showing that the Laplacian distribution provides a good fitting for the residual DCT coefficients, this section presents the Residual Statistical Modeling module which computes at both the encoder and decoder the Laplacian distribution parameter, α, that better fits the residual, R. The α parameter can be estimated using the maximum likelihood method [20] as in (46) where N represents the number of coefficients at the given DCT band for the full frame and R_k is the *k*th DCT coefficient value of that band:

$$\hat{\alpha} = \frac{N}{\sum_{k=1}^{N} |R_k|} \tag{46}$$

From (46), different α values are obtained for each residual DCT band but the same value for the same band in all the blocks of the full frame. However, to allow a better adaptation to the changing statistics within a DCT band, this means along space and time, it is proposed in this Thesis to estimate the α parameter at a (finer) DCT coefficient level, i.e. each residual DCT coefficient will have a different α value associated to it. In this case, adopting a finer modeling does not have any rate implications as no α parameter values have to be transmitted.

The proposed α parameter estimation at the DCT coefficient level is performed for each coefficient within each 4×4 (luma) block in a macroblock as follows:

- **Residual DCT coefficients inverse quantization –** First, the residual DCT coefficients are inverse quantized according to (44) where $P_r(u,v)$ and $P_q(u,v)$ are replaced by $R_r(u,v)$ and $R_q(u,v)$, respectively; the quantized residual DCT coefficients $R_q(u,v)$ correspond to the coefficients at the input/output of the entropy encoder/decoder.
- **DCT coefficients statistics accumulation –** According to (46), to estimate α it is first necessary to compute $\sum_{k=1}^{N} |R_k|$ for the relevant band. As has been seen in Section 2.4.1, the α estimation at the DCT coefficient level requires the computation of the average and variance of the full DCT coefficients band, which is available once the SI is created in the DVC scenario. To follow a similar strategy in predictive video coding, it would be needed to wait until the whole frame has been coded. Besides being time consuming, because the reconstruction operation (of all coefficients of all DCT bands) would have to be performed again with the proposed statistical reconstruction function, it would not allow to taking full advantage of the proposed reconstruction function, notably when Intra prediction is used. For these reasons, the proposed statistical reconstruction function is performed as the coding process evolves. In this context, the α parameter can only be estimated from R_k values meanwhile available for the current macroblock and all previously coded macroblocks. By considering all previously coded macroblocks in the α estimation process, it is likely to enclose R_k values whose order of magnitude is quite different from the one in the current macroblock (outlier residuals); note that outliers in statistical model parameter(s) estimation processes are in general responsible for lowering the statistical model accuracy and, therefore, should be avoided. For this reason, the α estimation technique proposed here considers only data meanwhile available for the current macroblock. Thus, $\sum_{k=1}^{N} |R_k|$ in (46) is

replaced by the sum of the absolute values of $R_r(u,v)$ for each DCT band (u,v) over the N_{prev_blocks} 4×4 blocks encoded/decoded so far (within a macroblock) according to:

$$V(u,v) = \sum_{k=1}^{N_{prev\_blocks}} \left| R_{r,k}(u,v) \right|. \quad (47)$$

- ***α* parameter estimation –** Finally, the α parameter for the (u,v) DCT coefficient is obtained from:

$$\hat{\alpha}(u,v) = \frac{N_{prev\_blocks}}{V(u,v)}. \quad (48)$$

Although sharing some similarity with (46), the α parameter estimation approach proposed in (48) allows a finer adaptation to the residual changing statistics within a band as each DCT coefficient has one α parameter associated to it. The proposed statistical reconstruction is expected to improve the predictive video coding reconstructed video quality (see in Chapter 1) and, therefore, it makes sense to apply it as the coding process evolves to allow creating better predictions. Note that when Intra 4×4 prediction is used, prior reconstructed samples in adjacent blocks are used to create the prediction for a given 4×4 block. To perform the DCT coefficients (statistical) reconstruction as the coding process evolves, it thus necessary to update $V(u,v)$ as the macroblock coding process takes place.

The $\hat{\alpha}(u,v)$ values obtained will be then used by the Statistical Reconstruction module to obtain the reconstructed DCT coefficients, as it will be seen in the next section.

4.3. Statistical Reconstruction

The main goal of the proposed Statistical Reconstruction module (which is present at both the encoder and decoder sides) is to reconstruct the DCT coefficients based on a statistical approach, differently from what is recommended by the H.264/AVC standard.

In this Thesis, it is proposed to replace the standard H.264/AVC inverse quantization (see Section 2.3.1) by the MMSE reconstruction function typically employed in DVC codecs [17] (see Section 2.4.2). By adopting a more adaptive approach, it is expected to improve the quality of the reconstructed video signal and, consequently, to improve the overall video codec RD performance. For that purpose, the Statistical Reconstruction module makes use of the inverse quantized prediction DCT coefficients, P_r, (obtained from (44)) and the inverse quantized residual DCT coefficients, R_r (obtained as described in Section 4.2.2).

The Statistical Reconstruction module includes two main steps, the DCT coefficients reconstruction bin bounds computation, and the DCT coefficients reconstruction, which will be described in detail in the following sections.

4.3.1. DCT Coefficients Reconstruction Bin Bounds Computation

The main objective of this step consists in determining the (lower and upper) bounds of the bin where each DCT coefficient should be reconstructed, hereafter called *DCT coefficients reconstruction bin,* needed for the statistical reconstruction function, as it will be seen in Section 4.3.2. It is important to remind that the statistical reconstruction function regards the DCT coefficients while the performed quantization regards the residual DCT coefficients. The DCT coefficient reconstruction bounds are obtained from the residual DCT coefficients reconstruction

bin bounds and the (reconstructed) prediction DCT coefficients since this data is available at both the encoder and decoder sides, thus allowing to guarantee the (needed) prediction synchronism. For this purpose, the following steps are performed to determine the DCT coefficients reconstruction bin bounds:

- **Residual DCT coefficients reconstruction bin width computation –** First, the width of the residual DCT coefficients reconstruction bin, i.e. the quantization step size $\Delta(u,v)$, is computed. Although the quantization step size can be computed from the quantization parameter sent in the bitstream, the resulting value will not be scaled to the order of magnitude of the DCT coefficients for which the reconstruction bin bounds have to be computed. Thus, it is proposed here to compute $\Delta(u,v)$ as the difference between the reconstructed (inverse quantized) values of any two adjacent residual DCT coefficients bins for a given band; note that the inverse quantization operation is performed as in (44).
- **Residual DCT coefficients reconstruction bin bounds computation –** Once Δ is known, the lower (L_R) and upper (U_R) bounds of the bin in which the residual DCT coefficient is reconstructed, hereafter called *Residual DCT coefficients reconstruction bin*, can be obtained. Note that the standard H.264/AVC solution reconstructs the residual DCT coefficients at a distance $f \times \Delta$ of the bin lower bound (see Section 2.3.1). Thus, L_R and U_R can be obtained from (49), (50) and (51), where sign{bin} corresponds to the signal operator defined in Section 4.1. As mentioned in Section 2.3.1, the dead-zone width of a NURQ is equal to $2\Delta(1-f)$. Thus, when the residual DCT coefficient reconstruction bin is 0, the lower and upper bounds, L_R and U_R, of that bin will both correspond to half of the dead-zone width, i.e. $2\Delta \times (1-f)/2$ (50); naturally, L_R and U_R will differ in their sign. In (49)-(51), the parameter f appears multiplied by 2^{15+Q_E} which corresponds to the scaling factor used in the regular H.264/AVC quantization process (see (42)). In case the residual DCT coefficient reconstruction bin is higher or lower than 0, the computation of L_R and U_R depends on Δ and the reconstructed value location within the bin; the reconstructed value is located at a distance $f \times \Delta$ of the bin lower bound whenever sign(bin) = 1 and at a distance $f \times \Delta$ of the bin upper bound whenever sign(bin) = -1, as illustrated in Figure 11 (see Chapter 2). Thus, considering that the residual DCT coefficient reconstruction bin is higher than 0, the bin lower bound is obtained by subtracting the distance $f \times \Delta$ to the reconstructed value $R_r(u,v)$ (51); U_R is then obtained as $L_R + \Delta$, since the bin width is Δ. A similar reasoning is applied when the residual DCT coefficient reconstruction bin is lower than 0:

$$\begin{cases} U_R(u,v) = R_r(u,v) + \dfrac{f \times 2^{15+Q_E}}{2^{15+Q_E}} \times \Delta \\ L_R(u,v) = U_R(u,v) - \Delta \end{cases}, \quad sign(bin) = -1 \tag{49}$$

$$\begin{cases} U_R(u,v) = 2 \times \Delta \times (1 - f \times 2^{15+Q_E})/2 \\ L_R(u,v) = -U_R(u,v) \end{cases}, \quad sign(bin) = 0 \tag{50}$$

$$\begin{cases} L_R(u,v) = R_r(u,v) - \dfrac{f \times 2^{15+Q_E}}{2^{15+Q_E}} \times \Delta \\ U_R(u,v) = L_R(u,v) + \Delta \end{cases}, \quad sign(bin) = 1 \tag{51}$$

- **DCT coefficients reconstruction bin bounds computation –** Finally, the lower (L) and upper (U) bounds of the bin where the DCT coefficient will be reconstructed are obtained from (52). Basically, by shifting the residual DCT coefficients reconstructed bin bounds

(previously computed) according to the reconstructed prediction value $P_r(u,v)$, the bin bounds where the (input video) DCT coefficient should be reconstructed can be found:

$$\begin{cases} L(u,v) = L_R(u,v) + P_r(u,v) \\ U(u,v) = U_R(u,v) + P_r(u,v) \end{cases} \tag{52}$$

4.3.2. DCT Coefficients Reconstruction

Finally, knowing the DCT coefficient bin bounds, the DCT coefficients are reconstructed using a statistical approach. The reconstruction function limits the error between the original frame and the reconstructed (decoded) frame to the quantizer coarseness since the reconstructed value has to be always within the DCT coefficients reconstruction bin bounds.

As mentioned in Chapter 2, the statistical reconstruction function, which is used in the context of the proposed video coding solution, is optimal in the sense that it minimizes the mean square error of the reconstructed value for each DCT coefficient and should enhance the standard H.264/AVC inverse quantization approach, which does not consider the motion compensated (or Intra) prediction (only the residual information). For this purpose, the Statistical Reconstruction module makes use of $P_r(u,v)$ obtained from (44), $\hat{\alpha}(u,v)$ obtained from (47), and $\Delta(u,v)$, $L(u,v)$ and $U(u,v)$ obtained as described in Section 4.3.1. Following the statistical reconstruction solution presented in Section 2.4.2 typically used in DVC [17], the reconstructed DCT coefficients, $X_r(u,v)$, are obtained from:

$$X_r(u,v) = \frac{\int_{L(u,v)}^{U(u,v)} xp(x|P_r(u,v))dx}{\int_{L(u,v)}^{U(u,v)} p(x|P_r(u,v))dx} \tag{53}$$

where $p(x|P_r)$ is the conditional probability density function modeling the correlation noise between the original DCT coefficients $X(u,v)$ and the 'side information' $P_r(u,v)$. This reconstructed value corresponds to the minimum mean-squared error estimate of the original DCT coefficients $X(u,v)$ [17]. As for DVC, $p(x|P_r)$ is a Laplacian distribution (see Section 4.2.1) and, thus, the following closed form can be derived from (53):

$$X_r(u,v) = \begin{cases} L(u,v) + \frac{1}{\alpha(u,v)} + \frac{\Delta(u,v)}{1 - e^{\alpha(u,v)\times\Delta(u,v)}} & , \quad P_r(u,v) < L(u,v) \\ P_r(u,v) + \frac{\left(\gamma + \frac{1}{\alpha(u,v)}\right) e^{-\alpha(u,v)\times\gamma} - (\delta + \frac{1}{\alpha(u,v)}) e^{\alpha(u,v)\times\delta}}{2 - (e^{-\alpha(u,v)\times\gamma} - e^{\alpha(u,v)\times\delta})} & , \quad P_r(u,v) \in [L(u,v), U(u,v)] \\ U(u,v) - \frac{1}{\alpha(u,v)} - \frac{\Delta(u,v)}{1 - e^{\alpha(u,v)\times\Delta(u,v)}} & , \quad P_r(u,v) \geq U(u,v) \end{cases} \tag{54}$$

which is used to avoid numerical computation of the integrals. In (13), $\delta = U(u,v) - P_r(u,v)$ and $\gamma = P_r(u,v) - L(u,v)$, as described in Section 2.4.2. As it can be observed from (54), the (statistically) reconstructed DCT coefficient value $X_r(u,v)$ depends on the prediction $P_r(u,v)$ location. Since the reconstructed value obtained from (54) corresponds to the minimum mean-squared error estimate of the original DCT coefficients $X(u,v)$, the statistical reconstruction approach allows improving the reconstructed video quality and the overall video codec RD performance, as it will be seen in the next chapter

Chapter 5

Predictive Video Coding with Statistical Reconstruction: Performance Assessment

The main target of this chapter is to present the performance evaluation of the video coding solution proposed in this Thesis, notably a H.264/AVC decoder with statistical reconstruction. To obtain a solid assessment, relevant and meaningful test conditions have to be defined, which are presented in Section 5.1. After, Section 5.2 will present the obtained results using relevant performance metrics in order meaningful conclusions may be derived.

5.1. Test Conditions

To evaluate the performance of the proposed video codec, notably in terms of RD performance, precise and representative test conditions must first be defined. In Section 5.1.1, the video sequences used to test the proposed video coding solution are presented alongside with their main characteristics. Next, Section 5.1.2 presents the coding conditions used to configure and control the H.264/AVC reference software while Section 5.1.3 finally presents the performance metrics.

5.1.1. Video Sequences

To evaluate the proposed solution, four video sequences have been selected, with different characteristics in terms of motion and texture, in order meaningful and representative results are obtained. The selected video sequences are *Night, City, Big Ships* and *Shuttle Start*. To have an idea on the content of each sequence, Figure 25 shows the first frame of each of the selected video sequences.

Figure 25 – First frame of the selected video sequences: top) Night (left) and City (right); bottom) Big Ships (left) and Shuttle Start (right)

All the sequences selected have rather distinct characteristics and can be classified in two categories, namely low and medium motion activity, as explained in the following:

- **Low motion activity:** The *Shuttle Start* and *Big Ships* sequences can be classified as low motion sequences due to the static camera and low object motion. *Shuttle Start* depicts the launch of a space shuttle at distance, thus giving the impression of a lower speed for the main object, the shuttle itself. The *Big Ships sequence* shows a big sailboat travelling on a river at average speed with a fast camera transition to another boat and the passengers aboard appearing at some stage.

- **Medium-High motion activity:** The *City* and *Night* sequences can be classified as medium motion sequences. The *City* sequence shows a downwards overview of a large city with skyscrapers; the sequence was clearly taken from a helicopter or plane in motion. The *Night* sequence depicts a busy street at night where cars are crossing the image field and people are crossing the road, walking and running.

In Table 3, the characteristics of each video sequence are presented, notably the spatial and temporal resolutions as well as the total number of frames coded for each sequence. For the case, HD sequences have been selected as this type of content was one of the main targets of the H.264/AVC standard.

Table 3 – Test video sequences characteristics

Motion Activity	Video Sequence	Spatial Resolution	Temporal Resolution [Hz]	Number of Frames
Low	Shuttle Start	1280×720	60	600
	Big Ships	1280×720	60	600
Medium	City	1280×720	60	600
	Night	1280×720	60	460

5.1.2. Coding Conditions

In this section, all the coding parameters and configurations used to evaluate the video codec proposed in this Thesis are presented. The total number of frames coded was 151, following the instructions in the VCEG document defining appropriate test conditions [21]. Although this document mentions that 150 frames should be coded for these specific sequences, the number used in this Thesis is 151 in order to enable a closed GOP with an Intra period of 15 frames. This solution allows to adjust the number of coded frames to the adopted GOP size. Moreover, the *Shuttle Start* sequence starts to be coded on frame number 150. Once again these choices were done following the instructions presented on the VCEG document previously mentioned.

The coding conditions used for the selected video sequences are presented in the following:

- **GOP Size –** The GOP used has M = 3 and N = 15, where M represents the distance between two anchor frames (I or P), thus two B frames in this case, and N represents the distance between two I frames. So the frames were organized in an IBBP prediction structure with the insertion of an I frame every fifteen frames.
- **Quantization Parameters –** Nine RD points were defined using the quantization parameter (Q_i) as presented in Table 4 and 5. To increase the RD performance, a cascading solution has been adopted for the quantization parameters, notably with the quantization parameter increasing one unit from the I to the P frames and another unit from the P to the B frames. Lower quantization parameters (and thus higher qualities) are thus used for the frames with longer prediction chains.

Table 4 – Quantization parameters used to define each RD point for the various video sequences (RD points 1-4)

Video Sequence	QP_1			QP_2			QP_3			QP_4		
	I	P	B	I	P	B	I	P	B	I	P	B
Night	10	11	12	12	13	14	14	15	16	18	19	20
City	10	11	12	12	13	14	14	15	16	18	19	20
Big Ships	10	11	12	12	13	14	14	15	16	18	19	20
Shuttle Start	10	11	12	12	13	14	14	15	16	18	19	20

Table 5 - Quantization parameters used to define each RD point for the various video sequences (RD points 5-9)

Video Sequence	QP_6			QP_6			QP_7			QP_8			QP_9		
	I	P	B	I	P	B	I	P	B	I	P	B	I	P	B
Night	24	25	26	30	31	32	36	37	38	42	43	44	48	49	50
City	24	25	26	30	31	32	36	37	38	42	43	44	48	49	50
Big Ships	24	25	26	30	31	32	36	37	38	42	43	44	48	49	50
Shuttle Start	24	25	26	30	31	32	36	37	38	42	43	44	48	49	50

- **Transform Size –** In the tests performed, only the 4×4 transform size has been used.
- **Number of Reference Frames** - The number of reference frames used for prediction has been 4.
- **Encoder Profile –** For all the tests, the High Profile has been used.
- **Other –** RD-Optimized mode decision and CABAC enabled.

5.1.3. Performance Evaluation Metrics

The quality metric used for the performance evaluation of the proposed solution in this Thesis is the PSNR. This metric is the most common used for video quality evaluation despite having some well known shortcomings in terms of expressing the perceptual (subjective) quality. The PSNR metric is a full reference metric as it measures the quality of the decoded frame with respect to the corresponding original frame as follows:

$$PSNR = 10log_{10}\left(\frac{{L_{max}}^2}{MSE}\right) \tag{55}$$

where L_{max} is the maximum luminance sample value (255 in this case as 8-bit samples are used), and MSE is the mean square error calculated between the decoded and corresponding original frames as:

$$MSE = \frac{1}{m \times n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[O(i,j) - D(i,j)]^2 \tag{56}$$

where $O(i,j)$ and $D(i,j)$ are the luminance values of the original and decoded frames, respectively, at position (i,j) and $m \times n$ represents the spatial resolution of the video sequence. Naturally, the average rate is computed by simply adding all bits used, dividing by the number of coded frames and multiplying by the frame rate.

Another metric largely used to evaluate the performance of one coding solution regarding another is the Bjontegaard metric [22]. The Bjontegaard metric allows to compute the PNSR average gain in dB or the average bitrate saving in percentage between two RD curves, typically using four RD points. While BD-rate expresses the average bitrate reduction of the assessed codec regarding the reference codec for a constant quality, BD-PSNR expresses the average PSNR increase of the assessed codec regarding the reference codec for a constant rate. This metric represents the average difference between the integrals of the two RD curves

(fitted with a parametric model using 4 data points) under comparison divided by the integration interval.

5.2. RD Performance Evaluation

The main objective for this section is to evaluate the overall RD performance of the proposed codec. Figure 26 to Figure 33 illustrate the RD performance of the proposed video coding solution when compared with the benchmark H.264/AVC video codec, corresponding to the H.264/AVC reference software version 18.2, and with the H.264/AVC codec with ARO as presented in Chapter 2; for better reading, the charts are divided for the lower and higher bitrates.

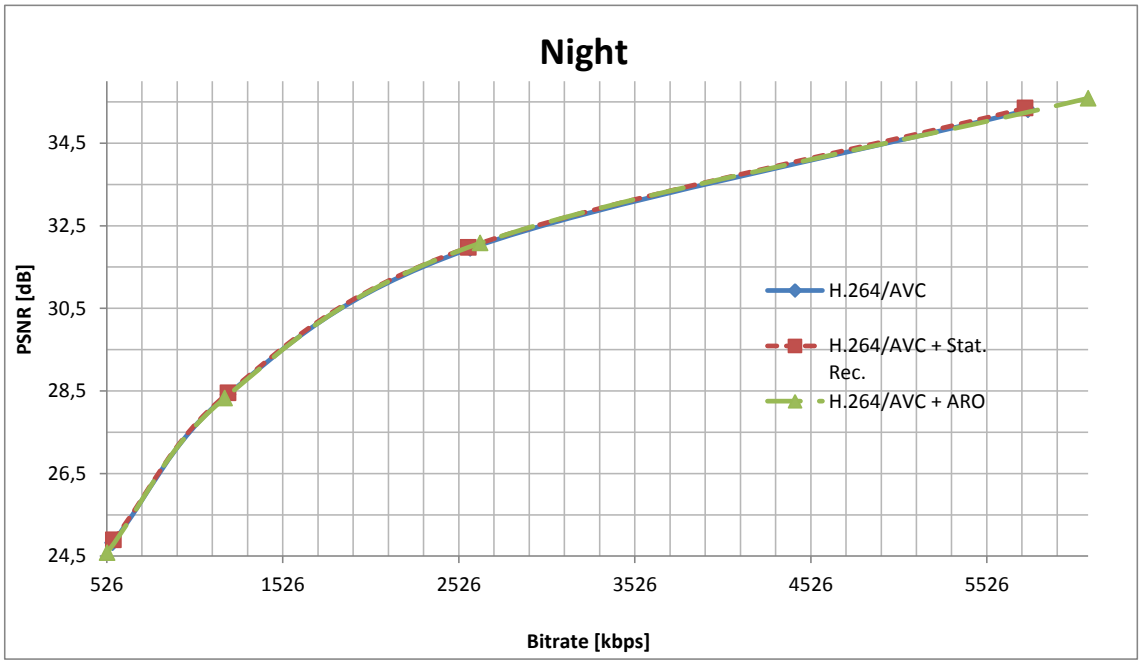


Figure 26 – RD performance comparison for the Night sequence: lower rates

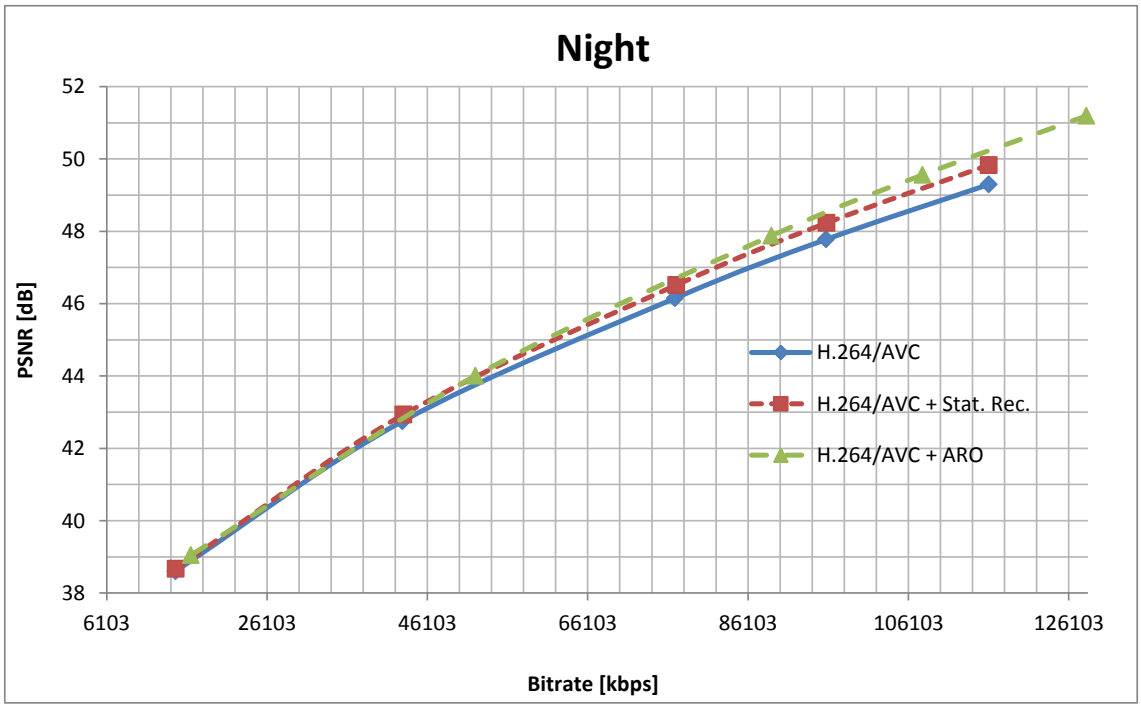


Figure 27 – RD performance comparison for the Night sequence: higher rates

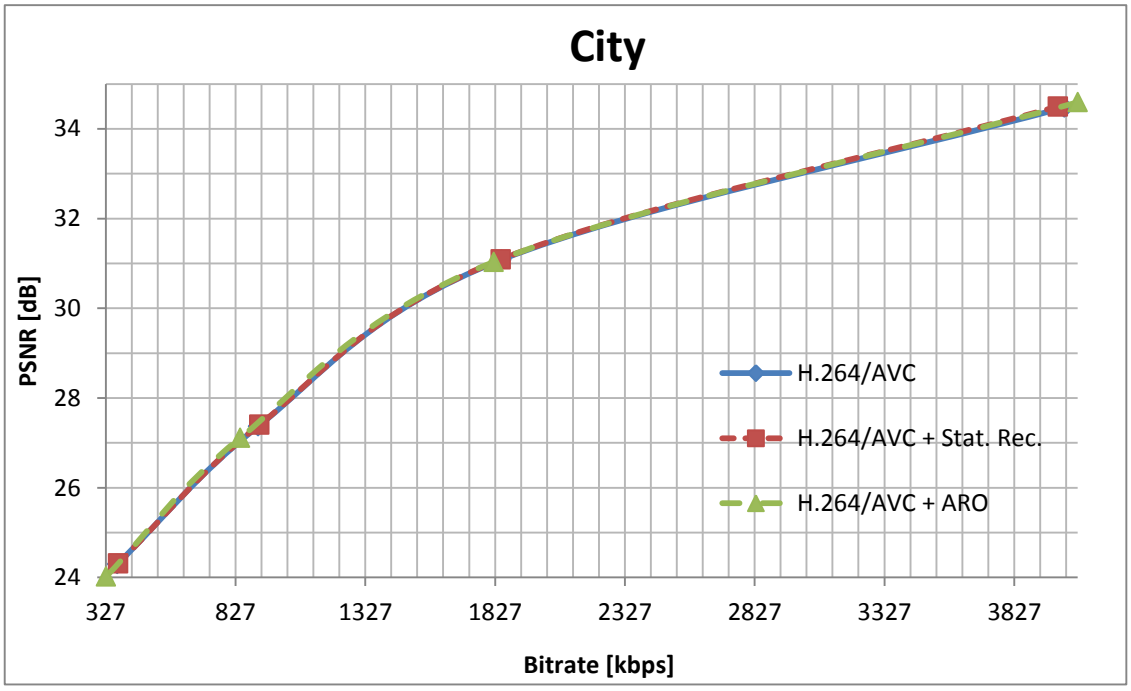


Figure 28 - RD performance comparison for the City sequence: lower rates

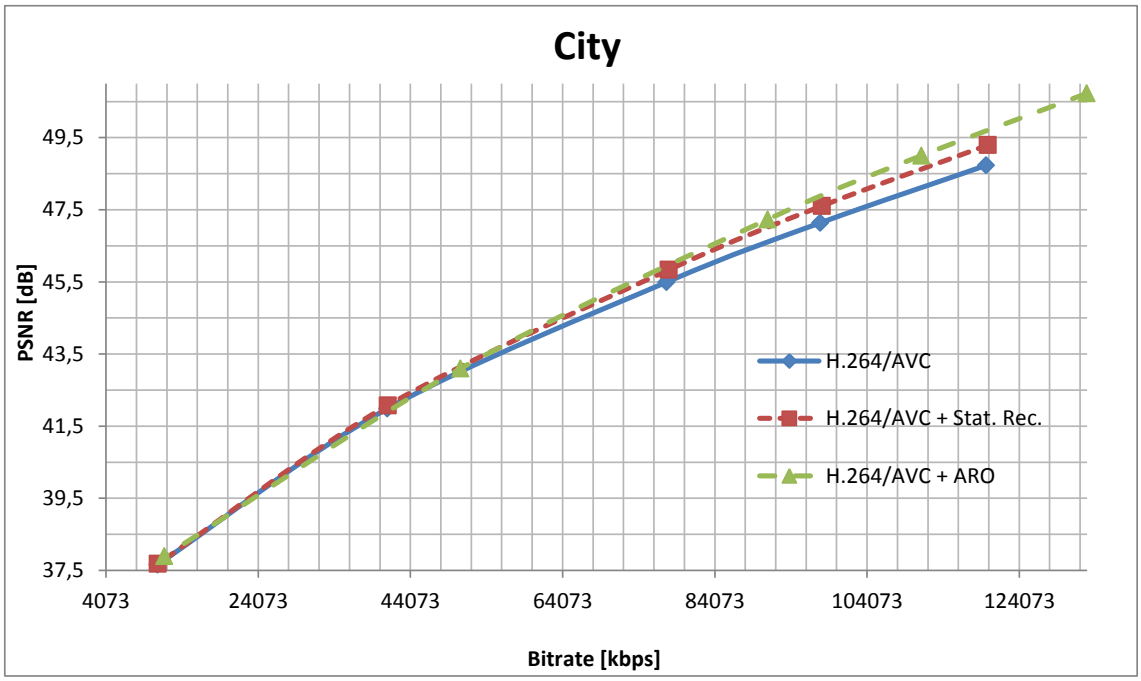


Figure 29 - RD performance comparison for the City sequence: higher rates

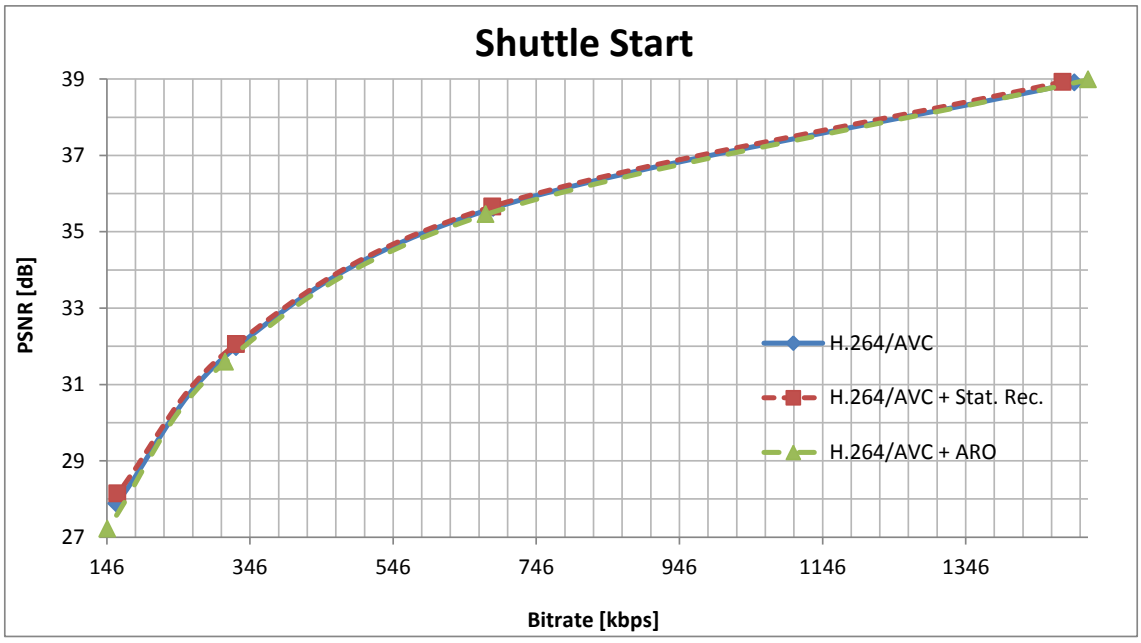


Figure 30 - RD performance comparison for the Shuttle Start sequence: lower rates

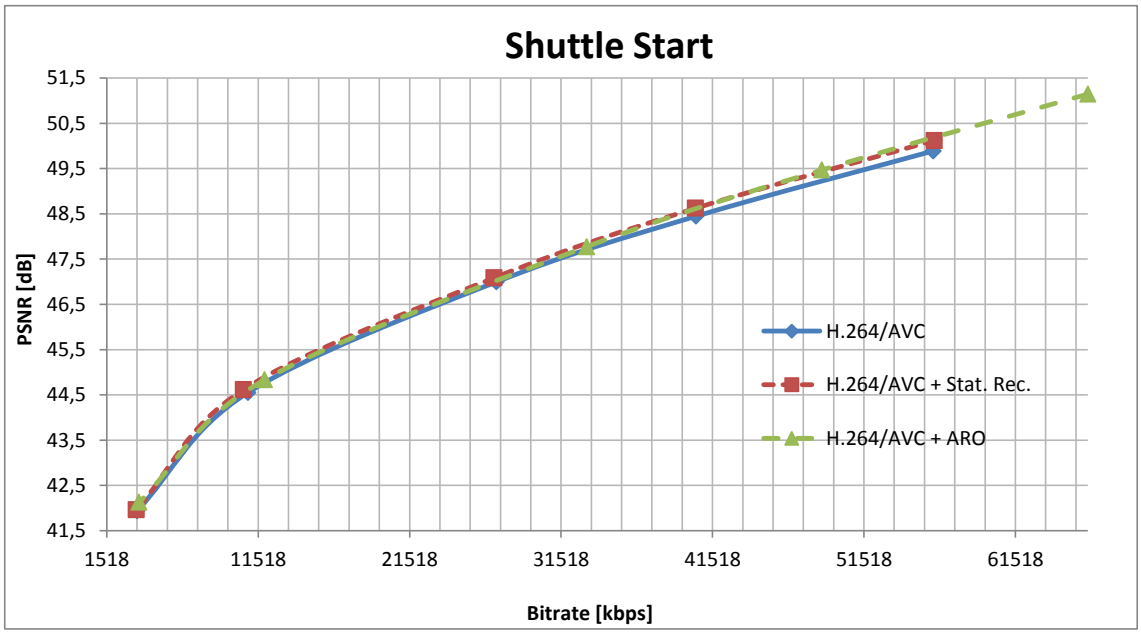


Figure 31 - RD performance comparison for the Shuttle Start sequence: higher rates

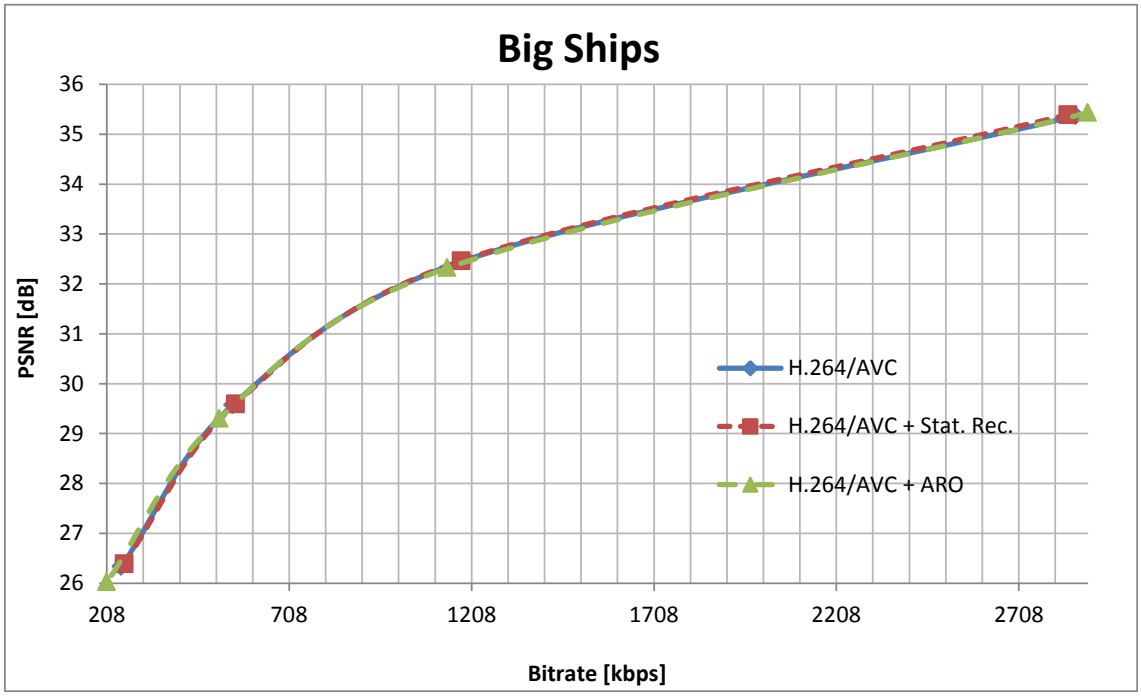


Figure 32 - RD performance comparison for the Big Ships sequence: lower rates

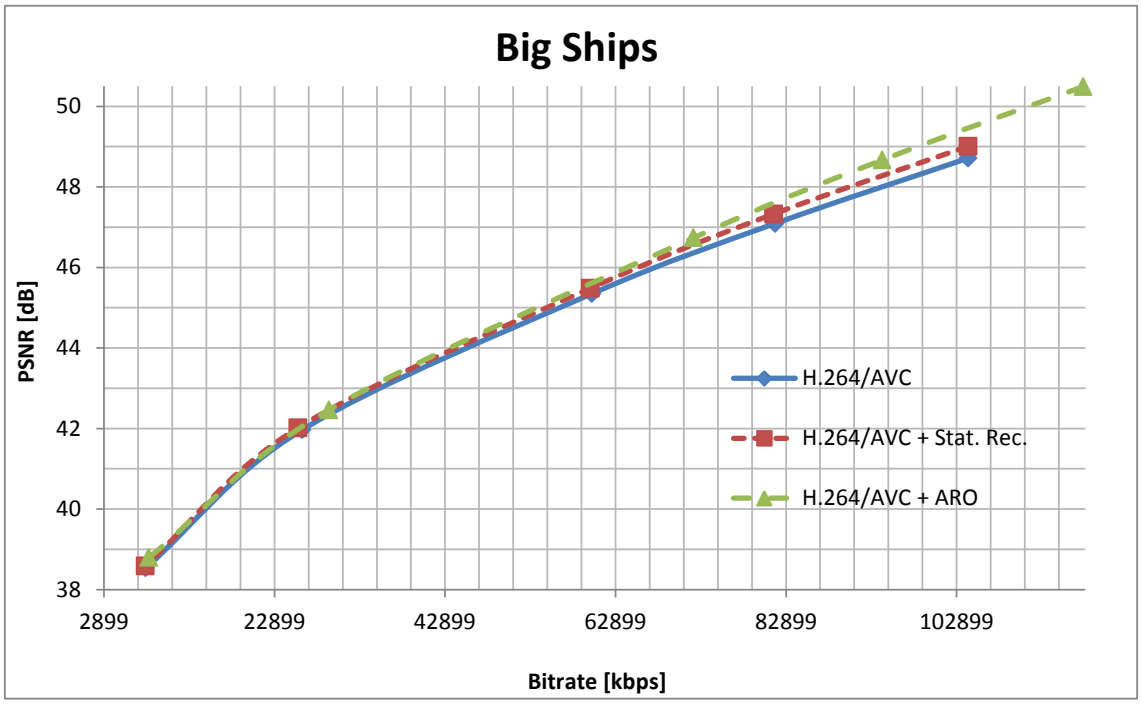


Figure 33 - RD performance comparison for the Big Ships sequence: higher rates

For a better understanding of the obtained results and a more precise comparison of the various coding solutions, also the Bjontegaard metric results are presented in Tables 6 to 9.

The results presented in Figure 26 to Figure 33 and Table 6 to Table 9 lead to the following conclusions:

- **H.264/AVC+Stat. Rec. versus H.264/AVC –** In Table 6, the results regarding the Bjontegaard metric for the higher rates considered in the tests performed are presented, making clear that the proposed solution outperforms the H.264/AVC codec for all of the video sequences tested. The proposed solution presents BD-Rate savings up to 4,71% and BD-PSNR gains up to 0,33 dB.

  Table 7 presents the results for the lower bitrates tested; although the gains are not as significant as for the higher rates, the proposed solution still manages to outperform the H.264/AVC codec for all of the video sequences tested, with BD-Rate savings up to 1,74% and BD-PSNR gains up to 0,09 dB. When analyzing the charts in Figure 26 to Figure 33, it is possible to observe PSNR gains up to 0.5 dB when looking for example at the *Night* or *City* video sequences, if points with the same bitrate are taken into account. Higher PSNR gains are observed for the more detailed sequences, such as *Night* and *City,* and higher bitrates (i.e. lower quantization step sizes), where it is more difficult for the rigid H.264/AVC reconstruction function, which reconstructs the residual DCT coefficients always at a fixed distance of the quantization bin lower bound, to minimize the (reconstructio*n) di*stor*tion;* in fact, for lower quantization step sizes, the reconstruction towards the center of the quantization bin is more beneficial for distortion *minimization* purposes [12]. By adjusting the reconstructed DCT coefficient value inside the quantization bin, the statistical reconstruction function is able to reduce the distortion, i.e. to increase the reconstructed

PSNR, for a similar bitrate. The lowest PSNR gain (about 0.2 dB) graphically observed is for the *Shuttle Start* sequence where a big portion of the frame area is still or smooth, which is typically associated to low residue values. Since low residue values are typically quantized to zero and both the statistical reconstruction and the H.264/AVC reconstruction functions output the prediction DCT coefficient for zero quantized residues, a lower PSNR gain is expected (when compared to the more detailed areas).

- **H.264/AVC+Stat. Rec. versus H.264/AVC+*ARO* –** When analyzing the results presented in Table 8, it is visible that the H.264/AVC+ARO solution has a better performance than the proposed codec both in terms of BD-Rate and consequently BD-PSNR. A particular case in this analysis is the *Shuttle Start* sequence where the proposed solution is able to outperform the H.264/AVC codec with ARO enabled, with BD-Rate savings of 0.91% and a marginal BD-PSNR increase of 0.03. As mentioned above, this sequence is characterized by a large portion of still or smooth areas within the video frames, which constrains the update of the rounding offset parameter to only small variations around an initial value (which corresponds to the value used in the H.264/AVC codec with ARO disabled). Thus, the ARO technique only allows to slightly improving the RD performance of the H.264/AVC codec while the proposed Statistical Reconstruction function, taking advantage of the non-still and detailed frame areas, is able to further improve the overall codec RD performance. In Table 9, the results for the lower rates tested are presented. It is possible to observe that for these lower rates the proposed video coding solution with statistical reconstruction manages to outperform the H.264/AVC+ARO codec for the *Shuttle Start* and *Night* video sequences. For the *Shuttle Start* sequence at lower rates, the proposed solution presents BD-Rate savings of 3,65% while having a BD-PSNR increase of 0.19 dB when compared with the H.264/AVC+ARO. From a graphical point of view, when comparing points with the same rate, the proposed video coding solution with statistical reconstruction performs in general better or similarly for all the test sequences at lower bitrates. However, at higher bitrates, the proposed video coding solution performs slightly worse than the H.264/AVC codec with ARO enabled when PSNR values are taken into account. This behavior may be explained by the fact that the ARO technique makes use of the original information (at the encoder) to dynamically update the rounding offset parameter during the encoding process, which does not happen in the statistical reconstruction solution.

Table 6 – Bjontegaard metric results for H.264/AVC + Statistical Reconstruction vs H.264/AVC: higher rates

Sequences	H.264/AVC (Reference codec)		H.264/AVC + Stat. Rec. (Proposed codec)		Bjontegaard Metric	
	Rate[kbps]	PSNR[dB]	Rate[kbps]	PSNR[dB]	BD-PSNR [dB]	BD-Rate [%]
Night	116141,65	49,30	116128,30	49,83	0,33	-4,71
	95843,61	42,78	95899,10	48,23		
	76996,82	46,15	77156,54	46,52		
	42,993,94	42,75	43116,40	42,93		
City	119723,08	48,73	119951,80	49,29	0,27	-4,19
	97912,42	47,14	98143,93	47,61		
	77718,95	45,49	78034,36	45,84		
	41023,032	41,98	41098,02	42,08		
Shuttle Start	56093,32	49,89	56156,79	50,12	0,12	-3,89
	40422,61	48,45	40400,48	48,62		
	27255,46	47,0	27095,45	47,08		
	10841,20	44,55	10537,78	44,62		
Big Ships	104309,03	48,72	104249,35	49,01	0,13	-3,09
	81639,89	47,09	81434,83	47,33		
	60170,03	45,35	59935,60	45,48		
	26150,79	41,97	25642,36	42,01		
Average Gains					**0,21**	**-3,95**

Table 7 – Bjontegaard metric results for H.264/AVC + Statistical Reconstruction vs H.264/AVC: lower rates

Sequences	H.264/AVC (Reference codec)		H.264/AVC + Stat. Rec. (Proposed codec)		Bjontegaard Metric	
	Rate[kbps]	PSNR[dB]	Rate[kbps]	PSNR[dB]	BD-PSNR [dB]	BD-Rate [%]
Night	5760,59	35,30	5744,19	35,35	0,04	-0,89
	2590,86	31,95	2580,94	31,98		
	1215,10	28,42	1215,56	28,45		
	558,88	24,82	563,77	24,89		
City	4019,71	34,48	3995,91	34,50	0,01	-0,32
	1851,61	31,09	1849,66	31,09		
	914,58	27,37	919,05	27,40		
	371,66	24,30	374,33	24,31		
Shuttle Start	1497,99	38,90	1481,28	38,93	0,09	-1,74
	685,82	35,62	684,59	35,66		
	326,54	31,97	326,59	32,06		
	159,33	27,88	160,76	28,15		
Big Ships	2863,85	35,37	2843,29	35,39	0,00	0,14
	1181,04	32,45	1179,51	32,47		
	553,42	29,57	560,85	29,59		
	247,77	26,33	255,36	26,39		
Average Gains					**0,03**	**-0,70**

Table 8 – Bjontegaard metric results for H.264/AVC + Statistical Reconstruction vs H.264/AVC + ARO: higher rates

Sequences	H.264/AVC + ARO (Reference codec)		H.264/AVC + Stat. Rec. (Proposed codec)		Bjontegaard Metric	
	Rate[kbps]	PSNR[dB]	Rate[kbps]	PSNR[dB]	BD-PSNR [dB]	BD-Rate [%]
Night	128316,86	51,19	116128,30	49,83	-0,22	2,87
	107867,63	49,56	95899,10	48,23		
	89016,30	47,88	77156,74	46,52		
	52078,95	44,01	43116,40	42,93		
City	132961,34	50,73	119951,80	49,29	-0,18	2,28
	111222,47	49,0	98143,93	47,61		
	91006,38	47,23	78034,36	45,84		
	50643,96	43,10	41098,02	42,08		
Shuttle Start	66305,62	51,15	56156,79	50,12	0,03	-0,91
	48737,80	49,47	40400,48	48,62		
	33204,92	47,77	27095,45	47,08		
	11944,58	48,83	10537,78	44,62		
Big Ships	117802,21	50,493	104249,35	49,01	-0,14	2,87
	94230,06	48,67	81434,83	47,33		
	72074,17	46,74	59935,60	45,48		
	29297,0	42,46	25642,36	42,014		
Average Gains					**-0,13**	**1.78**

Table 9 - Bjontegaard metric results for H.264/AVC + Statistical Reconstruction vs H.264/AVC + ARO: lower rates

Sequences	H.264/AVC + ARO (Reference codec)		H.264/AVC + Stat. Rec. (Proposed codec)		Bjontegaard Metric	
	Rate[kbps]	PSNR[dB]	Rate[kbps]	PSNR[dB]	BD-PSNR [dB]	BD-Rate [%]
Night	6101,99	35,59	5744,19	35,35	0,02	-0,56
	2647,93	32,08	2580,94	31,98		
	1196,05	28,33	1215,56	28,45		
	527,88	24,58	563,77	24,89		
City	4071,69	34,59	3995,91	34,50	-0,05	1,10
	1821,56	31,03	1849,66	31,09		
	845,09	27,11	919,05	27,40		
	328,11	24,02	374,33	24,31		
Shuttle Start	1516,99	38,98	1481,28	38,93	0,19	-3,65
	675,44	35,45	684,59	35,66		
	311,45	31,58	326,59	32,06		
	146,72	27,22	160,76	28,15		
Big Ships	2897,69	35,43	2843,29	35,39	-0,03	1,11
	1141,96	32,32	1179,51	32,47		
	517,03	29,30	560,85	29,59		
	208,46	26,02	255,36	26,39		
Average Gains					**0,13**	**-0,5**

This chapter aimed at evaluating the RD performance of the proposed statistical reconstruction solution. After defining the test conditions, the RD performance results as well as the Bjontegaard results were presented for the tested video sequences. The statistical reconstruction solution was compared with a standard H.264/AVC video codec with ARO enabled and disabled in order to fully understand its performance.

Chapter 6

Concluding and Future Work

In this chapter, the work presented in this Thesis is summarized and the main achievements identified. Moreover, some possible future work regarding the topics addressed is presented.

6.1. Summary

The first chapter of this Thesis introduces its context, highlighting the possibility of integrating coding tools used mainly in DVC paradigms in predictive video codecs such as the H.264/AVC standard. Moreover, this chapter clearly defines the objectives of this Thesis.

Chapter 2 focuses mainly in reviewing the relevant background technology for this Thesis, notably predictive video coding (with emphasis on the H.264/AVC video codec), distributed video coding, relevant aspects regarding quantization (notably the H.264/AVC quantization process and adaptive quantization algorithms) and, lastly, DVC correlation noise modeling and optimal reconstruction tools. This state-of-the-art review motivates the main target of this Thesis: to design a predictive H.264/AVC based video coding architecture where the usual inverse scalar quantization is replaced by a statistical reconstruction approach.

Following the background review, the architecture of the proposed video codec is presented in Chapter 3 alongside a brief description of the new modules introduced in the standard H.264/AVC codec architecture.

After introducing the designed codec architecture, Chapter 4 thoroughly presents the novel tools, emphasizing the importance and distinct functions of each of the modules, namely the *Ideal Transform Scaling and Quantization, Residual Statistical Modeling* and *Statistical Reconstruction* modules.

Finally, Chapter 5 assesses the proposed video coding solution. First, the test conditions are defined and after the RD performance is presented, notably using Bjontegaard metrics. The proposed codec is compared with the standard H.264/AVC codec with ARO disabled and enabled to understand the effective gains introduced by the video coding solution proposed in this Thesis.

6.2. Achievements

The main objective of the solution proposed in this Thesis is to improve the overall RD performance of the H.264/AVC video codec by adopting an alternative tool for the usual inverse scalar quantization.

To improve the codec performance, this Thesis proposes to adopt a DVC based statistical reconstruction method based on a correlation model for the H.264/AVC residual DCT coefficients. The Laplacian distribution has been adopted as the statistical model for the residual DCT coefficients, thus requiring estimating the Laplacian distribution parameter, α, for each residual DCT coefficient within a 4×4 luma block in a macroblock, using a maximum likelihood method.

By integrating the proposed technique, an improved codec was obtained and after assessed in comparison with the standard H.264/AVC video codec. Using the Bjontegaard metric to evaluate the RD performance of the proposed codec, it may be concluded that it outperforms the standard H.264/AVC codec for all of the video sequences tested, with BD-Rate savings up to 4,71% and a corresponding BD-PSNR increase up to 0.33 dB. When comparing the proposed codec with the H.264/AVC+ARO solution, the codec with statistical reconstruction has better performance for the lower rates evaluated, where BD-Rate savings up to 3,65% and a correspondent BD-PSNR increase of 0,19 dB were obtained. For the higher bitrates, the H.264/AVC+ARO solution has a better performance (except for the *Shuttle Start* sequence). Following the obtained results, it may be concluded that the objectives defined for this Thesis in Chapter 1 have been accomplished with success. As a consequence, a paper has been submitted to the IEEE Visual Communications and Image Processing (IEEE VCIP) Conference, to be held in Valletta, Malta, next December 2014.

6.3. Future Work

This section discusses some of the work that may be developed following the techniques proposed in this Thesis. The focus should be again obtaining a better RD performance, notably regarding the state-of-the-art video codecs. To address this challenge, the following research ideas can be explored:

- **ARO integration –** As concluded in Section 5.2, the H.264/AVC+ARO solution has some RD advantage over the proposed solution, mainly due to the fact that the ARO technique makes use of the original information (at the encoder), to dynamically update the rounding offset parameter during the encoding process. In this context, a possible integration of the proposed statistical reconstruction technique with the ARO algorithm may be beneficial as it has already been demonstrated that, for video sequences with still or smooth areas, the statistical reconstruction technique can outperform the ARO algorithm due to the offset parameter being constrained to only small variations around an initial value. So, the synergy between the two techniques may bring further RD gains by adopting a technique more adaptable to the video sequence characteristics.
- **Laplacian parameter refinement –** As discussed in Section 4.2.2, the Laplacian parameter α that characterizes the correlation noise model has a crucial role in the RD performance of the codec with statistical reconstruction. Thus, a possible future

development is to obtain even better α parameters to better fit the statisics of the source data, e.g. adopting a cumulative statistic not only at macroblock level such as performed in this Thesis.

- **HEVC extension –** Since the HEVC (High Efficiency Video Coding) standard has emerged recently to represent the state-of-the-art on video coding, the same idea proposed in this Thesis for the H.264/AVC standard may also be applied for the HEVC standard, namely to exploit the prediction using a statistical reconstruction to replace the inverse quantization.

In conclusion, the objectives proposed for this Thesis have been accomplished as it has been shown that replacing the usual inverse scalar quantization, as adopted in the H.264/AVC, with statistical reconstruction, thus combining the predictive and distributed video coding paradigms, can in fact lead to an improved RD performance.

References

1. T. Wiegand, G. J. Sullivan, G. Bjontegaard and A. Luthra, “Overview of the H.264/AVC Video Coding Standard”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, nº 7, pp. 560-576, Jul. 2003.
2. G. J. Sullivan, J.R. Ohm, W. J. Han, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) Standard”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, nº 12, pp. 1649-1668, Dec. 2012.
3. J. Ostermann, J. Borsmans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer and T. Wedi, (First Quarter 2004), “Video coding with H.264/AVC: Tools, Performance and Complexity (PDF), IEEE Circuits and Systems Magazine 4 (1), Retrieved January 31, 2011.
4. B. Girod, A. M. Aaron, S. Rane and D. R. Monedero, “Distributed Video Coding”, Proceedings of the IEEE, vol. 93, nº1, pp. 71-83, Jan. 2005.
5. G. J. Sullivan and T. Wiegand, “Video Compression – From Concepts to the H.264/AVC Standard”, Proceedings of the IEEE, vol. 93, nº1, pp. 18-31, Jan. 2005.
6. Fraunhofer – Heinrich Hertz Institute - http://www.hhi.fraunhofer.de, Mar. 2014.
7. F. Pereira. ‘Comunicação de Áudio e Vídeo’ course slides, Instituto Superior Técnico, 2012.
8. F. Pereira, “Comunicações Audiovisuais – Tecnologias, Normas e Aplicações”, IST Press, Lisboa, Portugal, 2009.
9. F. Pereira, C. Brites and J. Ascenso, “Distributed Source Coding: Theory, Algorithms and Applications”, Chapter 8, Academic Press, 2008.
10. Discover DVC Final Results - http://www.img.lx.it.pt/~discover/rd_performance.html, Mar. 2014.
11. I. E. Richardson, “The H.264 Advanced Video Compression Standard”, Wiley, Second Edition, 2010.
12. G. Sullivan, “Adaptive Quantization Encoding Technique using an Equal Expected-value Rule”, Doc. JVT-N011, Jan. 2005.
13. H. S. Malvar, A. Hallapuro, M. Karczewicz and L. Kerofsky, “Low-Complexity Transform and Quantization in H.264/AVC”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, nº 7, pp. 598-602, Jul. 2003.

14. Q. Xu, X. Lu, Y. Liu and C. Gomila, " A Fine Rate Algorithm with Adaptive Rounding Offsets (ARO)", IEEE Transactions on Circuits and Systems for Video Technology, vol. 19, nº 10, pp. 1424-1435, Oct. 2009.
15. X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, M. Ouaret, "The DISCOVER Codec: Architecture, Techniques and Evaluation", Picture Coding Symposium, Lisbon, Portugal, Nov. 2007.
16. C. Brites and F. Pereira, "Correlation Noise Modeling for Efficient Pixel and Transform Domain Wyner-Ziv Video Coding", IEEE Transactions on Circuits and Systems for Video Technology, vol.18, nº 18, pp. 1177-1190, Sep. 2008.
17. D. Kubasov, J. Nayak and C. Guillemot, "Optimal Reconstruction in Wyner-Ziv Video Coding with Multiple Side Information", Int. Workshop on Multimedia Signal Processing, Crete, Greece, Oct. 2007.
18. F. Bellifemine, A. Capellino, A. Chimienti, R. Picco, and R. Ponti, "Statistical analysis of the 2D-DCT coefficients of the differential signal for images," Signal Processing: Image Communication, vol. 4, no. 6, pp. 477–488, Nov. 1992.
19. G. S. Yovanof and S. Liu, "Statistical analysis of the DCT coefficients and their quantization error," Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, Nov. 1997.
20. E. Kreyszig, Advanced Engineering Mathematics, 7th ed., New York: Wiley, Chap. 25, pp. 1096-1099, 1993.
21. TK Tan, G. Sullivan, T. Wedi, "Recommended Simulation Conditions for Coding Efficiency Experiments Revision 2", ITU-T SC16/Q6 Doc. VCEG-AH10r3, 34th VCEG Meeting, Antalya, Turkey, Jan. 2008.
22. G.BjØntegaard, "Calculation of average PSNR differences between RD curves," Doc. VCEG-M33, Austin, TX, USA, Apr. 2001.

Annex A

Studying the Performance of the ARO Algorithm

In this appendix, some preliminary results regarding the performance of the ARO algorithm proposed in [12] and previously described in Section 2.3.3 are presented. First, the conditions in which the tests were performed are presented to allow the correct interpretation of the results and to enable performing a fair comparison with other results under the same conditions.

The experiments were conducted using various CIF and HD (720p) resolution sequences, with 150 frames for each sequence and a frame rate of 60 frames per second for all sequences. The tests were performed using the JM18.2 and JM9.6 H.264/AVC reference software encoder that represents the H.264/AVC codec at different stages of evolution, under the following conditions:

- IBBP with 150 frames encoded for each sequence;
- High Profile with Level IDC = 40;
- Motion Search Range = 64 for all sequences;
- Fixed QP, QP(I)=10, 14, 18, 22; QP(P)=QP(I)+1; QP(B)=QP(I)+2;
- Number of Reference Frames = 4;
- RD-Optimized mode decision enabled;
- CABAC;

The RD performance curves for each test sequence are presented in Figures 34 to 39. The charts include the RD performance with the ARO algorithm on and off for two versions of the H.264/AVC reference software (for the CIF sequences only). The results with ARO on and off allows to fully understand the impact of the ARO algorithm. From the results, it is clear that the RD performance with the ARO algorithm always outperforms the RD performance when ARO is turned off. This RD performance improvement is consistent for the sequences with different resolutions. The RD curves presented show that the gains are larger for the higher rates and

they may go up to 1-2 dB. The results obtained are consistent with those presented in Section 2.3.3 and extracted from [12].

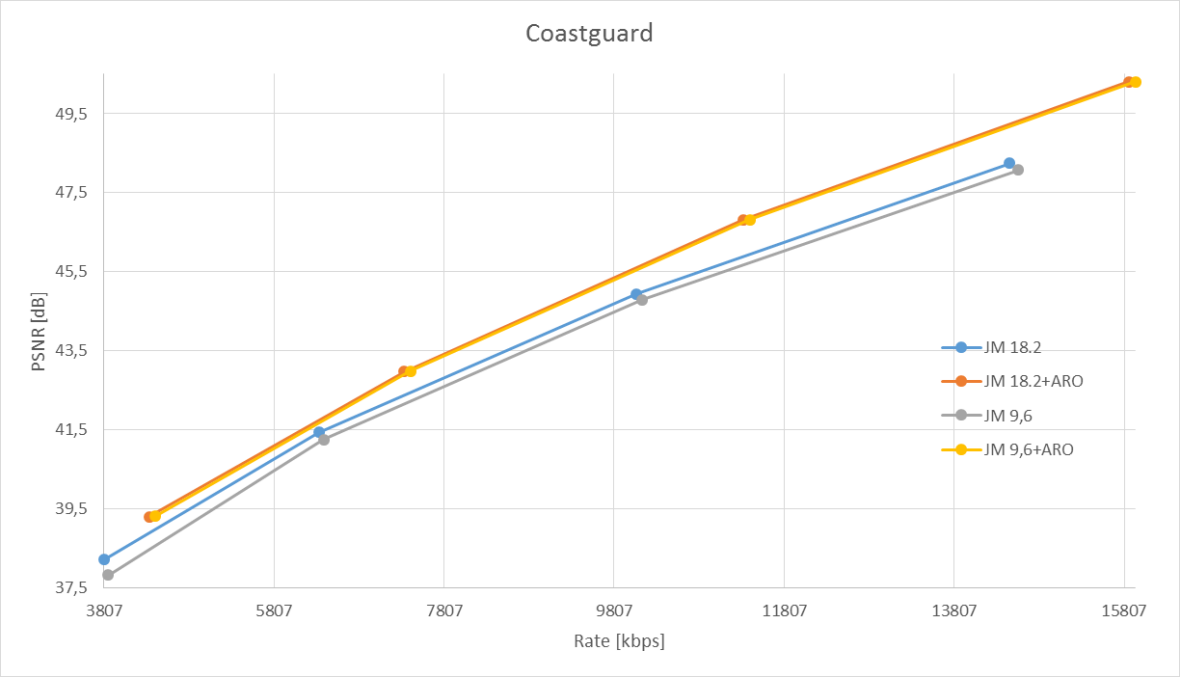


Figure 34 – RD performance comparison for the Coastguard CIF sequence

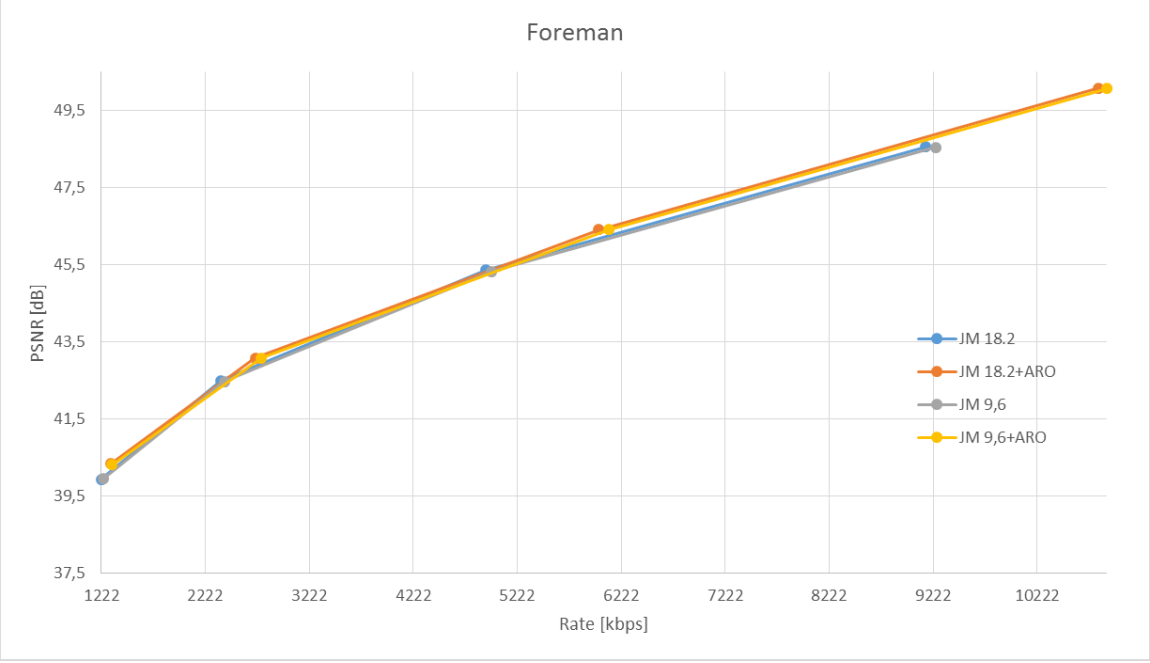


Figure 35 – RD performance comparison for the Foreman CIF sequence

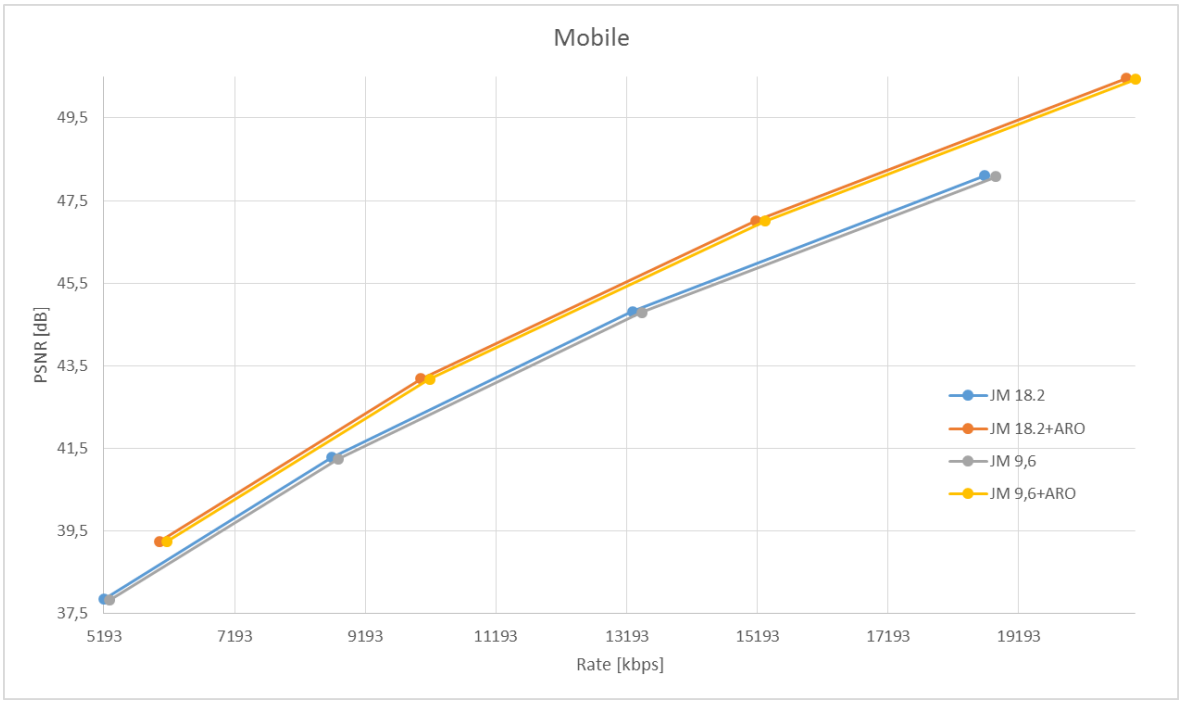


Figure 36 – RD performance comparison for the Mobile CIF sequence

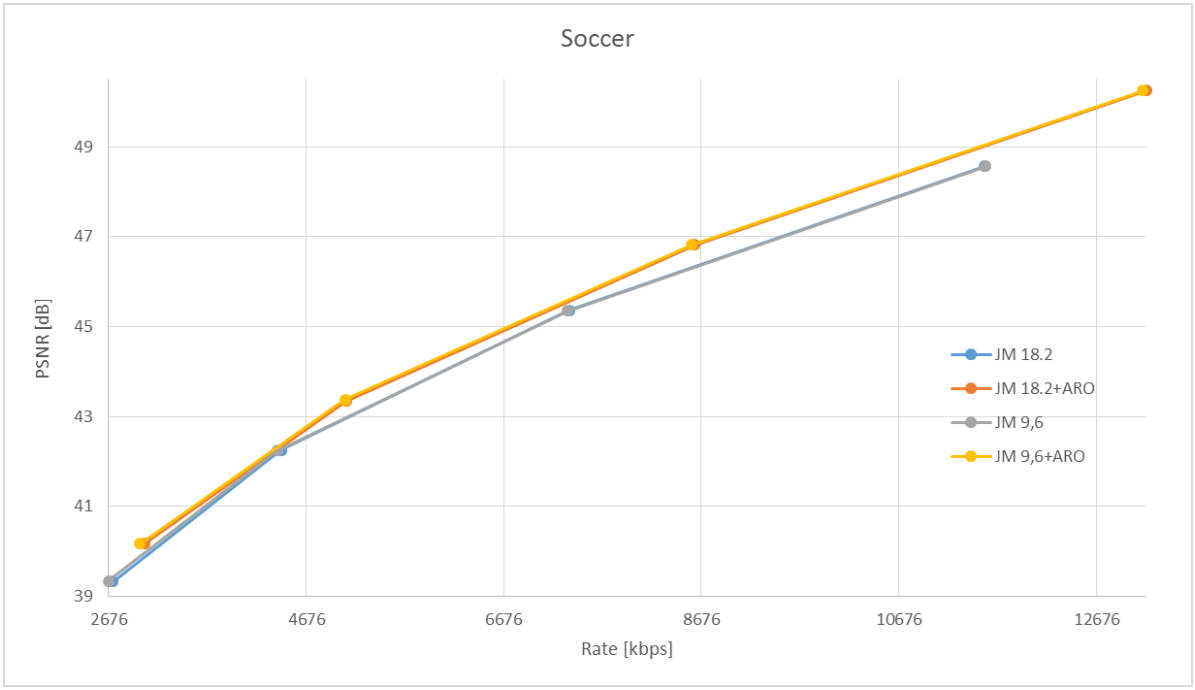


Figure 37 – RD performance comparison for the Soccer CIF sequence

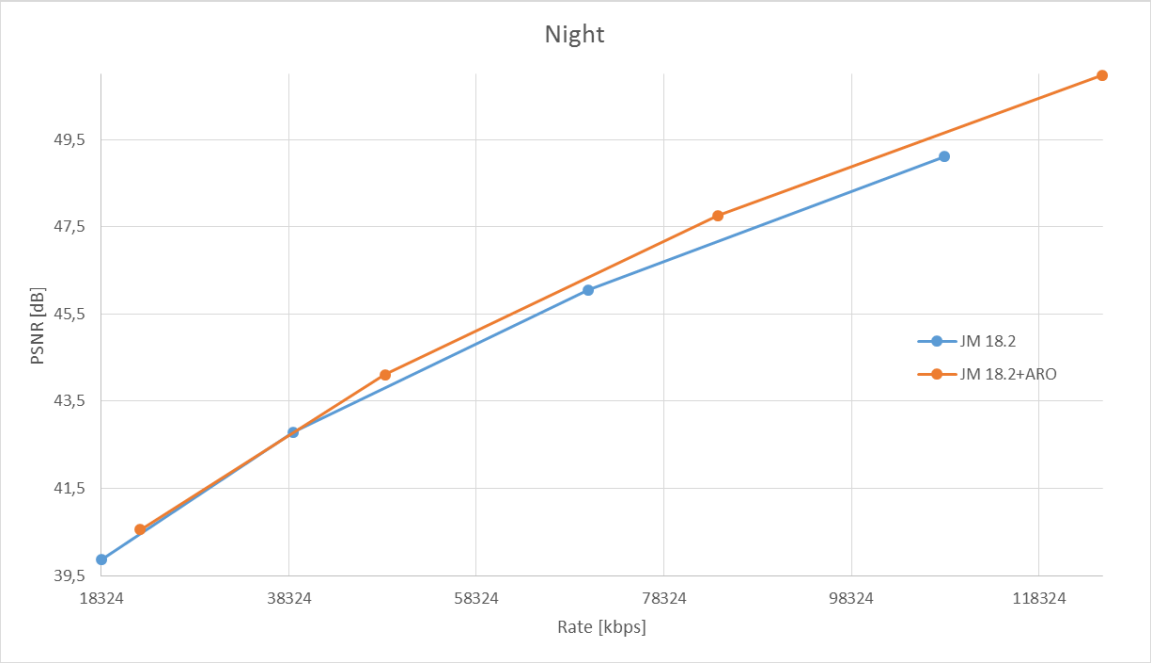


Figure 38 – RD performance comparison for the Night 720p sequence

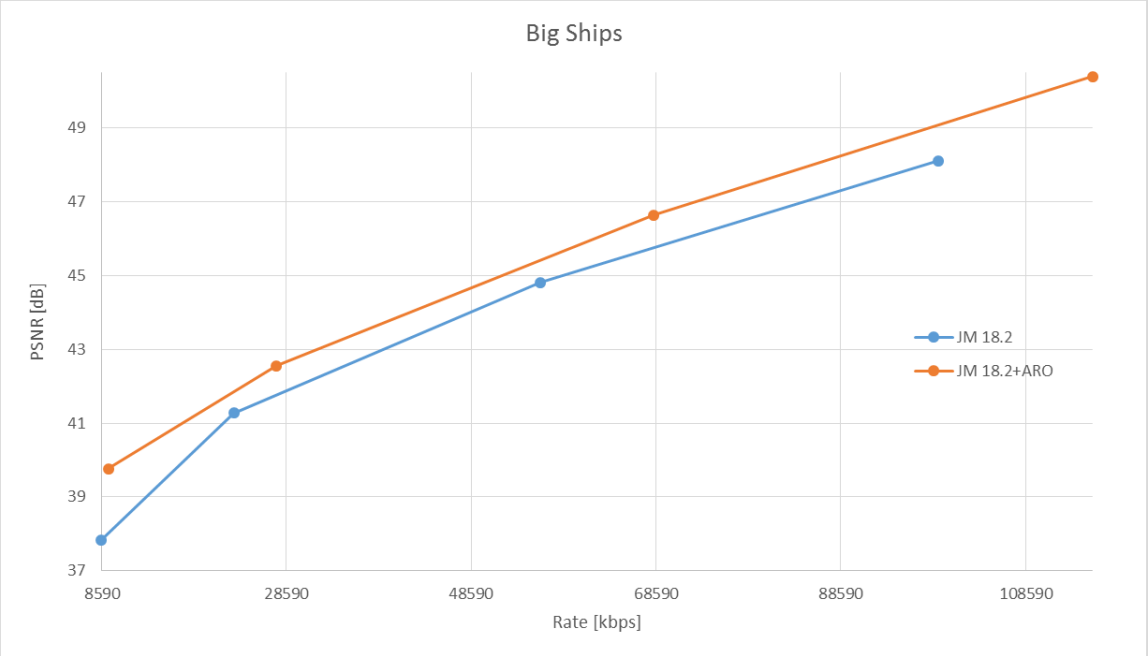


Figure 39 – RD performance comparison for the Big Ships 720p sequence