Dpto. de Teoría de la Señal y Comunicaciones
Universidad Carlos iii de Madrid

TESIS DOCTORAL

ALGORITHMS FOR COMPLEXITY MANAGEMENT IN VIDEO CODING

Autor: Amaya Jiménez Moreno
Directores: Dr. Fernando Díaz de María
Dr. Eduardo Martínez Enríquez

Leganés, 2016

Tesis doctoral:
Algorithms for Complexity Management in Video Coding

Autor:
Amaya Jiménez Moreno

Directores:
Dr. Fernando Díaz de María
Dr. Eduardo Martínez Enríquez

El tribunal nombrado para juzgar la tesis doctoral arriba citada, compuesto por los doctores:

Presidente:
Dr. Fernando Jaureguizar Núñez

Secretario:
Dr. Iván González Díaz

Vocal:
Dr. Javier Ruiz Hidalgo

acuerda otorgarle la calificación de:

Leganés, a

Detrás de los hastíos y los hondos pesares
Que abruman con su peso la neblinosa vida,
¡Feliz aquel que puede con brioso aleteo
Lanzarse hacia los campos luminosos y calmos!

Aquél cuyas ideas, cual si fueran alondras,
Levantan hacia el cielo matutino su vuelo
¡Que planea sobre todo, y sabe sin esfuerzo,
La lengua de las flores y de las cosas mudas!

Elevación - Charles Baudelaire

ABSTRACT

Nowadays, the applications based on video services are becoming very popular, e.g., the transmission of video sequences over the Internet or mobile networks, or the increasingly common use of the High Definition (HD) video signals in television or Blu-Ray systems. Thanks to this popularity of video services, video coding has become an essential tool to send and store digital video sequences.

The standardization organizations have developed several video coding standards, being the most recent H.264/AVC and HEVC. Both standards achieve great results compressing the video signal by virtue of a set of spatio-temporal predictive techniques. Nevertheless, the efficacy of these techniques comes in exchange for a high increase in the computational cost of the video coding process.

Due to the high complexity of these standards, a variety of algorithms attempting to control the computational burden of video coding have been developed. The goal of these algorithms is to control the coder complexity, using an specific amount of coding resources while keeping the coding efficiency as high as possible.

In this PhD Thesis, we propose two algorithms devoted to control the complexity of the H.264/AVC and HEVC standards. Relying on the statistical properties of the video sequences, we will demonstrate that the developed methods are able to control the computational burden avoiding relevant losses in coding efficiency. Moreover, our proposals are designed to adapt their behavior according to the video content, as well as to different target complexities.

The proposed methods have been thoroughly tested and compared with other state-of-the-art proposals for a variety of video resolutions, video sequences and coding configurations. The obtained results proved that our methods outperform other approaches and revealed that they are suitable for practical implementations of coding standards, where the computational complexity becomes a key feature for a proper design of the system.

RESUMEN

En la actualidad, la popularidad de las aplicaciones basadas en servicios de vídeo, como su transmisión sobre Internet o redes móviles, o el uso de la alta definición (HD) en sistemas de televisión o Blu-Ray, ha hecho que la codificación de vídeo se haya convertido en una herramienta imprescindible para poder transmitir y almacenar eficientemente secuencias de vídeo digitalizadas.

Los organismos de estandarización han desarrollado diversos estándares de codificación de vídeo, siendo los más recientes H.264/AVC y HEVC. Ambos consiguen excelentes resultados a la hora de comprimir señales de vídeo, gracias a una serie de técnicas predictivas espacio-temporales. Sin embargo, la eficacia de estas técnicas tiene como contrapartida un considerable aumento en el coste computacional del proceso de codificación.

Debido a la alta complejidad de estos estándares, se han desarrollado una gran cantidad de métodos para controlar el coste computacional del proceso de codificación. El objetivo de estos métodos es controlar la complejidad del codificador, utilizando para ello una cantidad de recursos específica mientras procuran maximizar la eficiencia del sistema.

En esta Tesis, se proponen dos algoritmos dedicados a controlar la complejidad de los estándares H.264/AVC y HEVC. Apoyándose en las propiedades estadísticas de las secuencias de vídeo, demostraremos que los métodos desarrollados son capaces de controlar la complejidad sin incurrir en graves pérdidas de eficiencia de codificación. Además, nuestras propuestas se han diseñado para adaptar su funcionamiento al contenido de la secuencia de vídeo, así como a diferentes complejidades objetivo.

Los métodos propuestos han sido ampliamente evaluados y comparados con otros sistemas del estado de la técnica, utilizando para ello una gran variedad de secuencias, resoluciones, y configuraciones de codificación, demostrando que alcanzan resultados superiores a los métodos con los que se han comparado. Adicionalmente, se ha puesto de manifiesto que resultan adecuados para implementaciones prácticas de los estándares de codificación, donde la complejidad computacional es un parámetro clave para el correcto diseño del sistema.

Contents

List of Figures

List of Tables

Acronyms

AVC	Advanced Video Coding
CABAC	Context-based Adaptive Binary Arithmetic Coding
CAVLC	Context-based Adaptive Variable Length Coding
CIF	Common Intermediate Format
CTB	Coding Tree Block
CU	Coding Unit
DCT	Discrete Cosine Transform
DPCM	Differential Pulse Code Modulation
DVB	Digital Video Broadcasting
HD	High Definition
HDTV	High Definition Television
HEVC	High Efficiency Video Coding
IDCT	Inverse Discrete Cosine Transform
ISO-MPEG	International Organization for Standardization-Moving Picture Experts Group
ITU-VCEG	International Telecommunication Union-Video Coding Experts Group
LRT	Likelihood Ratio Test

MB	Macroblock
MC	Motion Compensation
MD	Mode Decision
ME	Motion Estimation
MI	Mutual Information
MPEG	Moving Picture Experts Group
MV	Motion Vector
MVp	Motion Vector Predicted
PDF	Probability Density Function
PSNR	Peak Signal to Noise Ratio
PU	Prediction Unit
QCIF	Quarter-Common Intermediate Format
QP	Quantization Parameter
R-D	Rate-Distortion
RDO	Rate-Distortion Optimization
SAO	Sample Adaptive Offset
SD	Standard Definition
TU	Transform Unit
UHD	Ultra-High Definition
UHDTV	Ultra-High Definition TV
WPP	Wavefront Parallel Processing

Chapter 1

Introduction

In this PhD Thesis we address the problem of the high computational complexity of modern video coding standards. Specifically, we focus on the latest H.264/AVC and HEVC standards, which rely on a wide variety of coding options to obtain high compression ratios at the expense of a large computational complexity. We face this problem in the framework of the complexity control, where certain amount of computational resources are available for the coding process.

This chapter is organized as follows. First, we provide a brief description to the main concepts behind video coding in Section 1.1. The motivation of this Thesis is presented in Section 1.2. The control complexity problem is discussed in Section 1.3. The contributions of this Thesis are summarized in Section 1.4. Finally, the contents of the remainder of this Thesis are outlined in Section 1.5.

1.1 Video Coding

Nowadays, by virtue of the great development of technologies related to communications networks and processing power, multimedia applications are gaining preponderance. The majority of these applications can't be understood without a video component that must be compressed in order to be properly stored, transmitted, or manipulated.

Some of these applications can vary from digital television broadcasting, 3D movies and TV sets, DVD and Blu-Ray players, TV-on-demand and video streaming services over the Internet or mobile networks, cam-coders, PC-based editing systems, video-conference systems, etc. Furthermore, higher video resolutions are increasingly required. In fact, High Definition Television (HDTV) is today a very common resolution in consumer electronic, and even higher resolutions are beginning to be used, such as Ultra-High Definition TV (UHDTV) or Super Hi-Vision. Despite the growth in the capacity of communication networks and storage devices, networks and processors are not able yet to efficiently manage the large amounts of video signal data. For this reason, the video compression algorithms are currently a necessity.

During the last 30 years a huge work has been carried out in the development of video compression standards, facing the new requirements and promoting the interoperability among devices from different manufacturers, being both reasons essential to enable the global growth and success of the multimedia applications involving video data. International organizations, such as the ISO/IEC Moving Picture Experts Group (ISO-MPEG) and the ITU-T Video Coding Experts Group (ITU-VCEG), have been responsible for the development of standards for video compression. Both organizations jointly produced the H.264/MPEG-4 Advanced Video Coding (AVC) and the High Efficiency Video Coding (HEVC) standards, which have had a strong impact into all emerging applications, specially those involving increased video resolution, where these standards have provided notable improvements in terms of compression ratios. H.264/AVC and HEVC have incorporated more and more coding tools to further increase the compression ratios, but their complexity have also grown accordingly, being notably higher than that of previous standards and converting the video coding process into the bottleneck of many applications.

1.2 Motivation

The operation of a video compression system is based on removing redundancy in the spatial and temporal domains. Taking advantage of these redundancies, the video

compression systems significantly reduce the data required to efficiently represent a video sequence by means of differential coding techniques, which rely on coding the differences between the actual samples and the predicted ones based on these redundancies.

The H.264/AVC- and HEVC-based encoders rely on a huge set of coding options to build the predicted samples from the spatial and temporal redundancies. Every coding option results in a different prediction and generates a different Rate-Distortion (R-D) operation point (a different bit rate vs distortion trade-off) depending on the coding option and the video content. Among all these coding options the encoder has to select the optimal one to carry out the encoding process through an optimization method called *Rate-Distortion Optimization* (RDO).

The RDO process consists of minimizing the distortion subject to a rate constraint. This process is based on measurements of the actual rate and distortion values, which imply to carry out the complete coding and decoding processes for each coding option. To a large extent, the high computational complexity of these standards comes from this optimization method.

The motivation of this work is to address the resulting high computational complexity of the two latest video coding standards. This becomes essential for any practical implementation of a video compression system. For example, to adapt the system complexity to the available computational resources of a portable device, or to accommodate the compression process to the network capabilities for video transmission. For this purpose, we aim to develop complexity control methods able to smartly reduce the number of available coding options, alleviating the computational burden of the video compression system with slight losses in the compression efficiency.

1.3 Complexity Control in Video Coding

There are two approaches to manage the complexity of a video coding system: complexity reduction methods [Tourapis and Tourapis, 2003, Zhu et al., 2002, Zhang et al., 2003, Choi et al., 2003, Li et al., 2005, Gonzalez-Diaz and Diaz-de Maria,

2008, Grecos and Mingyuan, 2006, You et al., 2006, Kuo and Chan, 2006, Saha et al., 2007, Zhou et al., 2009, Martinez-Enriquez et al., 2007, Martinez-Enriquez et al., 2009, Martinez-Enriquez et al., 2010, Martinez-Enriquez et al., 2011, Lu and Martin, 2013, Kim et al., 2014, Yusuf et al., 2014, Leng et al., 2011, Shen et al., 2012, Shen et al., 2013, Xiong et al., 2014, Choi et al., 2011, Tan et al., 2012, Zhang et al., 2013, Ahn et al., 2015, Lee et al., 2015] and complexity control methods [Ates and Altunbasak, 2008, Gao et al., 2010, Kannangara et al., 2008, Huijbers et al., 2011, Vanam et al., 2007, Vanam et al., 2009, Su et al., 2009, Tan et al., 2010, Kannangara et al., 2009, da Fonseca and de Queiroz, 2009, da Fonseca and de Queiroz, 2011, Li et al., 2014, Correa et al., 2011, Correa et al., 2013, Ukhanova et al., 2013, Grellert et al., 2013].

The complexity reduction methods are quite common techniques. They are designed to reduce as much as possible the computational burden of the video compression systems, maintaining a reasonably good performance in terms of compression ratio. However, the results of the complexity reduction methods depend heavily on the coding system configuration and the video content and, therefore, these techniques are not capable of guaranteeing that the complexity is kept around a given target.

The strength of the complexity control methods is their ability to solve this problem adapting its performance to the available resources for the coding process. The goal of a complexity control method is to reduce the number of available coding options to be used by the video compression system in some smart way, which allows for meeting certain target complexity according to the system capabilities while maximizing the video coding performance.

These techniques are not so common as the those aiming at complexity reduction. In fact, previous complexity management efforts in the literature still suffer several drawbacks, e.g., some works are unable to adapt its performance to time varying conditions in either complexity requirements or video content (e.g., [Tourapis and Tourapis, 2003, Zhu et al., 2002, Choi et al., 2003, Leng et al., 2011, Choi et al., 2011, Shen et al., 2012]; in fact, these are actually complexity reduction methods);

other works require complex *off-line* trainings that hamper the generalization ability necessary to serve to the wide range of visual features of video sequences (e.g., [Gao et al., 2010, Kannangara et al., 2009, Correa et al., 2011]); in other cases, the coding options are reduced without accounting for the potential impact on the performance (e.g., [Vanam et al., 2007, Grellert et al., 2013, da Fonseca and de Queiroz, 2011]); finally, some methods are unable to manage different video resolutions (e.g., [Tan et al., 2010, Su et al., 2009, Kannangara et al., 2008, Ukhanova et al., 2013]).

Furthermore, the control of the computational complexity of a video coding system faces a number of difficulties: the requirements of each application are far from those of others; when the number of coding options allowed by the algorithm is too much narrowed, the system efficiency may be compromised, incurring large losses in compression efficiency; etc. To properly address these difficulties, is critical to design a complexity control method able to manage all kind of requirements while achieving a high compression efficiency.

1.4 Contributions

In this Thesis we propose two complexity control methods within the framework of the two latest video compression standards, H.264/AVC and HEVC, since both are widely used nowadays. These proposed methods solve some drawbacks of the state-of-the-art proposals, in such a way that they are capable of serving to a wider variety of real video applications.

To achieve a proper design, the latest video coding standards have been analyzed in detail, identifying the contribution of each coding tool to both the compression efficiency and the complexity of the system. Thus, a detailed statistical analysis has been carried out to gain an insight into the behavior of the compression systems for different types of sequences, covering a wide range of contents, characteristics and resolutions.

It should be noted that, depending on the characteristics of the sequence, some coding options become more efficient than others. Thus, the statistical analysis

has allowed us to guide the design of our complexity control system, so that it is capable of selecting the most adequate coding options for every sequence, system capabilities, and application, solving the main problem of several state-of-the-art proposals (e.g., [Vanam et al., 2007, Grellert et al., 2013, da Fonseca and de Queiroz, 2011]).

Another contribution of this Thesis, likely the most relevant, is the capability of adapting the methods over time. The variability inherent to video sequences and the potential changes in the available processing or communication resources undoubtedly demand adaptive methods. Otherwise, the developed methods would fail when changes happened in the video content or in the requirements of the applications (e.g., [Gao et al., 2010, Kannangara et al., 2009, Correa et al., 2011]).

Moreover, a complexity-aware design of our methods has allowed us to incorporate them to the considered standards with a negligible computational impact, solving the problem of complex designs (e.g., [Vanam et al., 2009, Ukhanova et al., 2013]).

Furthermore, the proposed methods have been thoroughly tested over a wide variety of resolutions, overcoming a problem of several proposals in the state-of-the-art (e.g., [Tan et al., 2010, Su et al., 2009, Kannangara et al., 2008, Ukhanova et al., 2013]).

Finally, we have compared our proposals with other approaches in the state-of-the-art with excellent results. In a few words, we have obtained adequate complexity reductions without incurring significant losses in compression ratio or quality.

Summarizing, our main contributions are:

1. In-depth analysis of complexity-related issues in H.264/AVC and HEVC:
    - Statistical analysis of the main compression tools in terms of coding performance and complexity for a wide range of contents.
2. Design of novel complexity control methods for H.264/AVC and HEVC with the following strengths:
    - Use of statistical methods to select the most adequate coding option for any video content, system capabilities, and application.

- *On the fly* adaptation to varying contents or networks capabilities.
- Comprehensive experimental validation in terms of coding complexity and efficiency.
- Notable performance improvement when compared with other approaches in the state-of-the-art.
- Managing a large variety of resolutions.
- Entailing a very low operational load.

1.5 Thesis Outline

The remainder of this Thesis is organized as follows. First, in Chapter 2, we provide a basic introduction to the main concepts in video coding. Then, in Chapter 3, we explain the H.264/AVC and HEVC standards, focusing on the specific coding tools that are relevant to our work. In Chapters 4 and 5, we describe in detail the proposed complexity control algorithms for H.264/AVC and HEVC standards, respectively. Finally, in Chapter 6, we present our conclusions and discuss future lines of research.

Chapter 2

A Basic Introduction to Video Coding

2.1 Introduction

The video coding systems have become an essential part of modern devices and applications. The goal of these systems is to reduce the amount of digital data necessary to represent a video sequence. Any coding system requires two devices: a compressor (encoder) and a decompressor (decoder). The first is designed to reduce the amount of data to represent a video sequence (coding process). The second is devoted to invert the coding process and recover the video signal (decoding process). Figure 2.1 summarizes these processes.

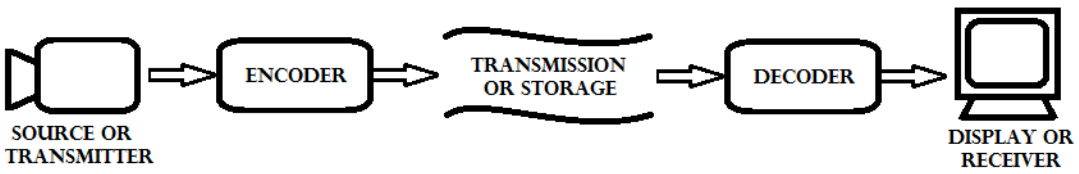


Figure 2.1: Coding and decoding processes.

The operation of a video coding system is based on removing redundancy in the spatial and temporal domains. In Figure 2.2 an example of two consecutive video frames is shown. In the highlighted region of the first frame the spatial redundancy becomes evident, as the content variations within the bounding box are very low. If

we pay attention to both frames, we can see that the differences between them are very small, which denotes high temporal redundancy. In Figure 2.2(c) the difference between both frames is shown; as it can be seen, there are many values close to zero and only small differences appear in the edges of some objects. If we take into account that a video sequence normally consists of 25 or 30 frames per second, we can expect a strong temporal redundancy.

(a) Frame #1. A region exhibiting spatial redundancy is highlighted.

(b) Frame #2.

(c) Difference between both frames.

Figure 2.2: Two consecutive video frames of *Container* sequence at CIF resolution.

In this chapter we describe the main concepts and tools developed in video coding to reduce the amount of data based on these features, focusing on the concepts that will be required to understand the proposed methods on this Thesis. First, a general description of the digitization process of a video sequence is presented. This is necessary to understand the nature of the data that the coding system receives. Then, the general procedures used by the video coding system are explained in detail.

2.2 Digital Video Representation

To obtain a proper digital representation of a video signal, a sampling process in the temporal and spatial dimensions must be carried out. In the temporal domain, video frames will be captured at regularly spaced time instants. In the spatial domain, every frame is horizontally and vertically sampled in picture elements (pixels) to

represent the scene inside the frame. Moreover, to digitize a color video sequence, three components must be considered. Then, every pixel value must be quantized, i.e., the actual number is approximated by one value taken from a discrete set of values. The digital representation of any signal entails certain losses due to the sampling and quantization processes involved in the analog to digital conversion. However, when this process is properly carried out, the losses are negligible from the human perception point of view.

2.2.1 Temporal and spatial resolutions

To convert the original video signal into digital data, it should be sampled in both the temporal and spatial domains (see Figure 2.3).

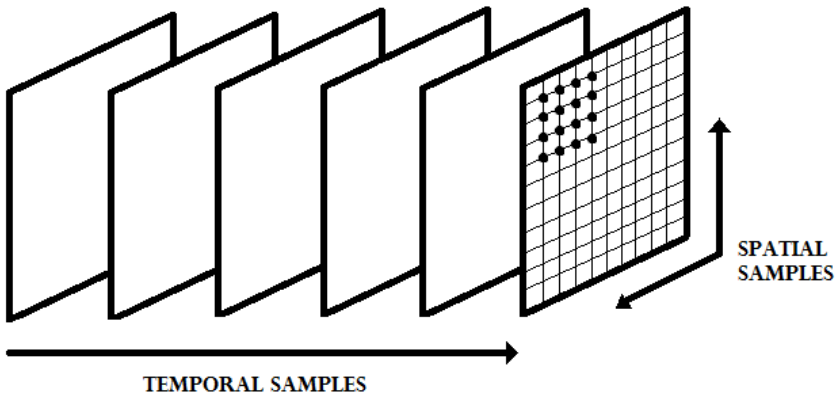


Figure 2.3: Sampling in temporal and spatial domains of a video sequence.

Typical temporal resolutions are 25 or 30 frames per second, although the HD systems may use up to 60 or 120 frames per second. This parameter defines the quality with which the movement is perceived: the higher the frame rate, the better the movement representation. Regarding the spatial sampling, there are many different standard resolutions, summarized in Table 2.1. The spatial resolution can vary from a very low value of 176×144 pixels, adequate only for little devices with a very limited processor, up to HD (1280×720), UHD (3840×2160), or Super Hi-Vision (7680×4320), suitable for wide screens, professional applications, or high-end devices.

Video format	Frame size in pixels (width×height)
QCIF (Quarter-CIF)	176×144
CIF (Common Intermediate Format)	352×288
SD (Standard Definition)	704×576
HD (720)	1280×720
1080	1920×1080
2k	2048×1536
UHD (4k)	3840×2160
4000p	4096×3072
Super Hi-Vision (8k)	7680×4320

Table 2.1: Common spatial resolutions.

Just considering the resulting number of pixels per second, it becomes obvious that digital video exhibits high redundancy in both temporal and spatial domains, as it was illustrated in Figure 2.2.

2.3 Video Coding

2.3.1 The hybrid DPCM/DCT model

The main video coding standards are based on a generic model consisting of two main blocks: the first, known as DPCM (Differential Pulse Code Modulation), aims to obtain a prediction of the signal to then compute the difference (residue) between the original and the predicted data; the second consists of a transform and an entropy coder. Usually, the complete system is referred to as hybrid encoder DPCM/DCT, as the Discrete Cosine Transform (DCT) is the most widely used transform.

The main advantage of coding the residue instead of the original signal is that, as long as the prediction is good, the residue will have lower energy than the original signal and, consequently, will be more efficiently encoded.

In Figures 2.4 and 2.5 we show the block diagram of a hybrid encoder and decoder based on DPCM/DCT, respectively. Basically, these figures summarize the coding and decoding processes of a frame F_n of a video sequence. In the encoding process, the encoder produces a bit-stream that contains the compressed binary representation of the frame F_n. In the decoding process, from this bit-stream, an approximate version

of the original signal is obtained F'_n. Usually, F'_n is different from F_n, since the encoding process entails certain losses (quantization). In the next subsections both processes are explained in detail.

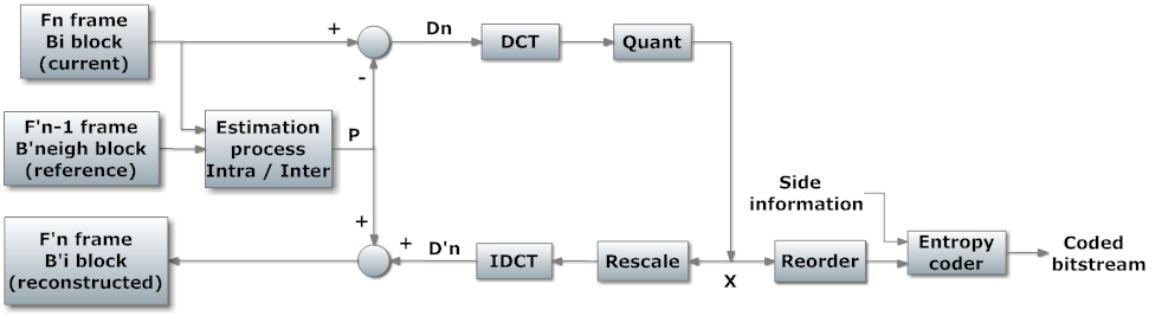


Figure 2.4: DPCM/DCT hybrid video encoder.

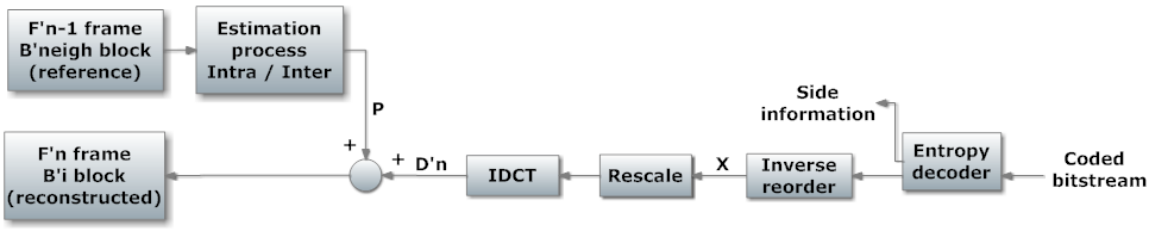


Figure 2.5: DPCM/DCT hybrid video decoder.

2.3.2 The encoding process of the hybrid DPCM/DCT model

As it can be seen in Figure 2.4, there are two data flows in the encoder block diagram: from left to right, the encoding data flow, and, from right to left, the reconstruction process. Let's begin with the encoding data flow:

1. The frame to be coded F_n is divided into smaller work units. These work units are blocks of pixels B_i that, depending on the standard considered, will have different sizes.

2. Each block goes through a prediction process. In this stage, taking advantage of the video redundancies, the encoder obtains a prediction P of the current block. This prediction may be based on the neighborhood of the current block B'_{neigh}

(taking advantage of the spatial redundancy), what it is called Intra prediction, or may be based on the previous coded frames F'_{n-1} (exploiting the temporal redundancy), what it is called Inter prediction. Both types of predictions will be explained with more detail later.

3. Once the prediction of the current block is obtained, the residue D_n between the predicted and the original blocks is calculated.

4. The residue D_n is then transformed. In the two standards considered in this Thesis, the DCT is used to transform each residue block into another block (of the same size) in the transformed domain, where the information is organized according to its frequency content. Usually, the coefficients with the highest energy correspond with the low frequencies, while is common to find close to zero coefficients in high frequencies. In such a way, the DCT is able to compact the information in a few (low frequency) coefficients. An example of the DCT of an image block is shown in Figure 2.6. As it can be seen, the low frequency coefficients (located in the top left corner) present higher energy and, as we move away towards higher frequencies, we see lower energy.

(a) Original block. (b) DCT coefficients block.

Figure 2.6: Example of DCT transform of a 64×64 pixel block.

5. The next stage is the quantization (Quant in Figure 2.4). The DCT-coefficient block is quantized to produce a new block denoted as X. The quantization process approximates the original values of the DCT coefficients by means of a set of discrete amplitudes. This stage is responsible for the coding losses, since

once the coefficients are quantified, the original values could not be recovered anymore. An uniform quantizer, which is a kind of quantizer very similar to those used in H.264/AVC and HEVC, is illustrated in Figure 2.7.

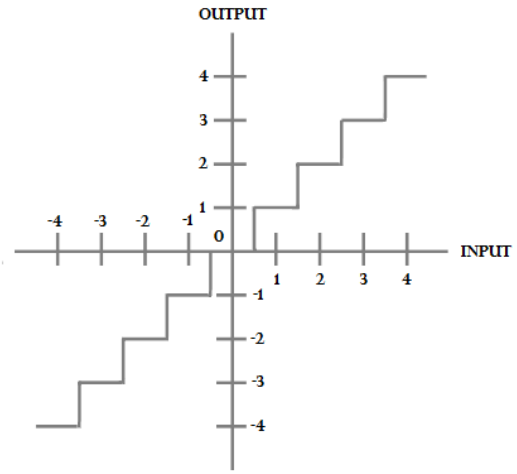


Figure 2.7: Uniform quantizer.

6. Subsequently, the block of quantified DCT coefficients is scanned in the order shown in Figure 2.8 (zig-zag scanning). Thanks to this reordering, most of non-zero coefficients are grouped together at the beginning, favoring the appearance of long runs of zero coefficients when the higher frequencies are scanned. The entropy coders are able to encode more efficiently this kind of data exhibiting long zero runs.

7. Finally, the quantified and reordered DCT coefficients go through the entropy coder to obtain the final compressed bit-stream that represents the frame F_n. The entropy coder also encodes some side information related to the way the prediction is built along with some headers.

Once the encoding data flow has been explained, we will go through the reconstruction path of the video encoder. The reason to include this reconstruction path is that the predictions built by the encoder must be based on information previously

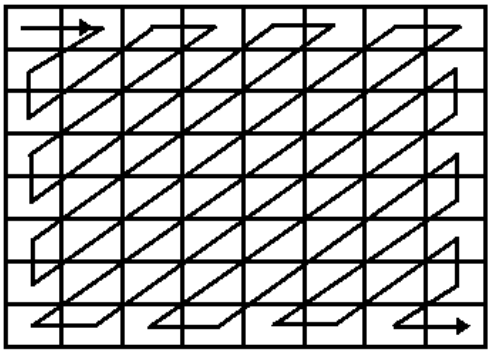


Figure 2.8: Zig-zag scan order.

decoded, obtaining in this way the same reconstructed values as the decoder. Otherwise, the decoder would incur additional errors because it would be unable to build the same prediction than the encoder.

The reconstruction data flow follows these steps:

1. At the input to this reconstruction path, we have the quantized DCT coefficients X. The X values are rescaled, which is the inverse quantization process, obtaining the reconstructed DCT coefficients, after the quantization losses.

2. Then, the reconstructed DCT coefficients go through an inverse DCT (IDCT) process, to obtain a spatial-domain version of the reconstructed residue block D'_n.

3. The reconstructed residue D'_n is added to the prediction P, calculated in previous stages, obtaining the reconstructed block B'_i. All these blocks B'_i will form part of the reconstructed frame F'_n. It must be noted that this frame F'_n is a lossy version of the original F_n, but it is exactly the same that the one obtained by the decoder, whenever there are no transmission errors.

2.3.3 The decoding process of the hybrid DPCM/DCT model

The basic steps followed by the decoder are summarized next:

1. The decoder receives the bit-stream generated by the encoder and it is entropy decoded to obtain the quantized DCT coefficients and the side information (headers and data to correctly build the prediction from previously decoded data). From here, the processes followed to encode the information must be inverted.

2. The coefficients are read following the inverse order to that shown in Figure 2.8, obtaining the block of quantified DCT coefficients (X in Figure 2.5), with the lower frequencies in the top left corner and higher frequencies appearing as we move away from that corner.

3. The remaining process is the same that we have described for the reconstruction data flow of the encoder. After the rescaling, we get the decoded DCT coefficients.

4. The IDCT is applied to the decoded DCT coefficients to obtain the decoded residue D'_n.

5. Relying on the side information of the bit-stream, the decoder can build the prediction P from previously decoded data.

6. The decoded residue D'_n is added to the prediction P, obtaining the decoded block B'_i that will form part of the decoded frame F'_n.

2.3.4 The spatial and temporal predictions

Before explaining the specific coding tools defined in the standards considered in this Thesis, we provide some insight into the procedures to construct a spatial or temporal prediction. In this Section we present a summary of the main concepts and procedures used in both kinds of predictions.

2.3.4.1 Spatial prediction

The prediction based on spatial redundancy is called Intra prediction. To obtain a prediction using the spatial redundancy, it is necessary to use previously coded

samples in the same frame and in the neighborhood of the current block. As the encoder scans the image on a block-by-block basis from left to right and from top to bottom, the neighborhood consists of the blocks at the left, top, and upper left positions relative to the block to be coded (as these blocks are already coded and adjacent). It is important to remember that the Intra prediction must be build from reconstructed pixels so that the encoder produces the same prediction as the decoder.

It should be noted that, depending on the considered video coding standard, there are several ways to combine the information of the surrounding pixels to build the Intra prediction. All these ways will be explained in the Chapter 3, devoted to describe the coding tools of both H.264/AVC and HEVC.

In Figure 2.9 a simple example of how to combine the information of the surrounding pixels (in gray color) is presented. Specifically, an average of the pixels previously coded is used to build the prediction for a 2×2-pixel block (in white color) and the residue block is obtained from this prediction.

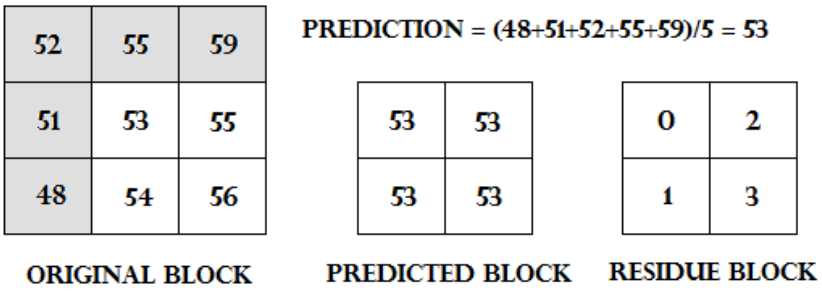


Figure 2.9: Example of Intra prediction for a 2×2-pixel block.

If the Intra prediction is properly calculated, the residue block generated by this process will have lower energy than the original block and, since lower energy signals can be coded with fewer bits, the compression process will be more effective. In the example presented in Figure 2.9, if the original pixel values were coded, 6 bits would be required to represent this information. However, the residue information would need only 2 bits. Therefore, we would require a total of 8 bits to represent the information, instead of the 24 bits required by the original block.

The Intra prediction is very useful in video coding standards, e.g., the first frame of any video sequence must be coded using Intra prediction, as there are no previous frames to use temporal prediction. Moreover, whenever we find a block with the pixels values equal to or very similar to the neighboring blocks (e.g., in an homogeneous area), the Intra prediction will allow a very efficient compression of that block.

2.3.4.2 Temporal prediction

The process dedicated to build predictions using the temporal redundancy is called Inter prediction. This is one of the most powerful tools used by the encoder to achieve high compression ratios.

To build Inter predictions the encoder follows two stages: first, the motion estimation (ME) and, second, the motion compensation (MC).

In the ME process, for each block to be coded, a search in previously coded frames is carried out to find the most similar block. In this way, the encoder minimizes the energy of the residue, which results in fewer bits to encode the block. An example of a ME process is shown in Figure 2.10.

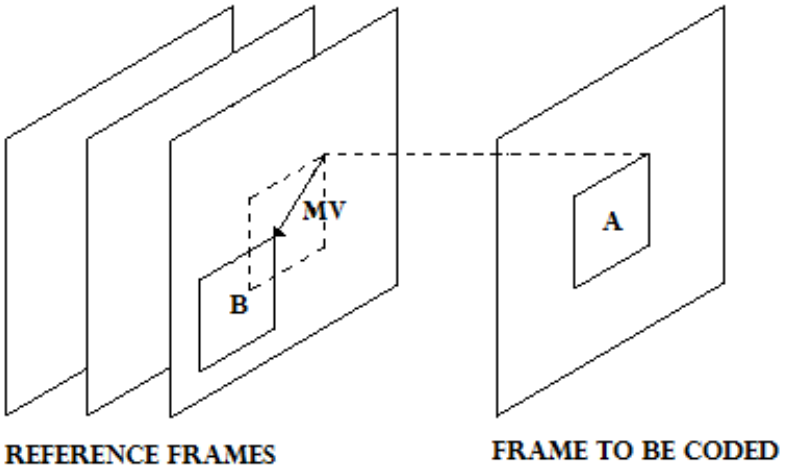


Figure 2.10: Example of Inter prediction. Motion estimation process. B is the best match for A. MV is the motion vector relating the location of B with respect to A.

The search process can be carried out in several frames, called reference frames, to achieve higher accuracy, selecting the best match between original and predicted blocks. The reference frames must be reconstructed frames to guarantee that the decoder produces exactly the same prediction. Once the position with the best match has been found, it is stored as a Motion Vector (MV), which indicates the location of the best match relative to the location of the block to be coded. The MC stage uses the MV previously found to retrieve the best match block and to build the prediction that will be subtracted from the block to be coded, obtaining the residue. The MV and an index to identify the reference frame must be send to the decoder so that it can produce the same prediction.

Thanks to the temporal prediction the number of bits required to represent a video sequence can be reduced very efficiently. The main problem associated with the Inter prediction is the computational burden required to carried it out.

2.3.4.3 Types of coded frames

Depending on the available predictions in every frame, there are several classes of coded frames that can be used in the encoding process. Typically, there are three types of frames: Intra (I), Predicted (P), and Bipredicted (B) frames.

The I-frames are those composed only for Intra-predicted blocks. Normally, these kind of frames are used to code the first frame of a video sequence, where temporal prediction is not feasible. Moreover, they are also useful to avoid the propagation of errors and for issues related to random access (allowing the decoding process to start at different points of the bit-stream); for these reasons, I-frames are usually periodically inserted in the coded sequence.

The P-frames are pictures where the Inter prediction is also available, i.e., both kinds of predictions are allowed and the encoder is responsible for selecting the best prediction for each block (this process will be explained later). In the P-frames the Inter prediction is carried out from previously coded frames that, in the display order, are frames from past time, i.e., the coding and display order are the same. Figure 2.11 shows a simple example of a sequence coded with P-frames. This coding pattern is

called IP, where the first frame is coded as I and the remainder are coded as P (until the appearance of other periodic I frame). The numbers in the brackets indicate the coding order and, as it can be seen, display and coding orders are the same. Since the P-frames need information of previous frames, they can not be used for random access.

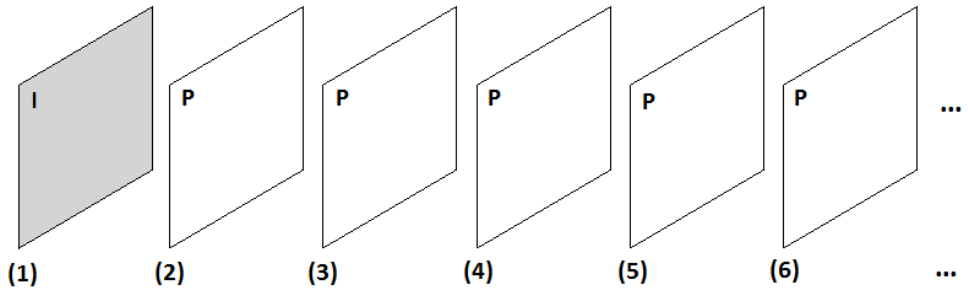


Figure 2.11: Example of IP coding pattern.

Both, I- and P-frames can be used as reference frames for other frames in the ME process. However, if there is some error in the decoding of the P-frames used as references, this error will be propagated. The I-frames are able to stop this error propagation problem.

The B-frames are pictures for which Inter prediction is also available, but, in this case, each block can be predicted from one or two reference frames, taking into account one frame from the past and one from the future. As the B-frames are not used in this Thesis, we do not provide more details about them. The interested reader is referred to [Richardson, 2003] for more information.

Chapter 3

Video Coding Standards

3.1 Introduction

This chapter is devoted to the explanation of H.264/AVC and HEVC standards, as both are considered in this Thesis. We begin with a brief historical introduction to the development of video coding standards. Next, the H.264/AVC and HEVC standards will be explained in detail, focusing on the coding processes more related to the problem addressed in this Thesis.

3.2 A brief history of video coding standards

The H.261 recommendation of the ITU-VCEG standardization organization can be considered the first video coding standard. Its final version was published in 1993 [ITU-T, 1993]. The target applications of this standard were the videophone and the video-conference over ISDN networks with rates of p×64 kbps, where p takes the values 1 to 30. This was the first standard with a hybrid DPCM/DCT block-based structure. It was able to achieve bit rates down to 40 kbps, outperforming previous efforts to compress video data.

Shortly after, also in 1993, ISO-MPEG published its first standard, the MPEG-1 [ISO, 1993]. In this case, the target applications were those related to the digital

video storage and transmission of standard definition television, with bit rates up to 1.5 Mbps. MPEG-1 was also based on the hybrid DPCM/DCT model and it already involved I-, P-, and B-frames, achieving good compression ratios.

Both organizations, ISO-MPEG and ITU-VCEG, worked together in the standard H.262/MPEG-2 [ISO/IEC, 1995] (1995). This standard outperformed the previous ones defining new tools such as interlaced and scalable video coding, achieving higher quality and compression ratios. The applications of this standard were mainly oriented to storage, e.g., the DVD player, and to broadcast TV (it defined a video transport specification that is used still today in Digital Video Broadcasting (DVB)). All these reasons made MPEG-2 one of the most successful video coding standards ever developed.

Then, the ITU-VCEG organization focused on the low bit rates applications and developed the H.263 standard [ITU-T, 1998] (1998). It was based on H.261, but with many more coding tools available, thanks to which it achieved improved compression ratios and flexibility.

Moreover, ISO-MPEG also published in 1998 the MPEG-4 Part 2 standard [ISO, 1998], oriented to what was called the new generation video applications. It was able to manage real world video sequences and computer generated graphics, and gave support to object based video coding, in which each object in a scene could be independently coded. It was oriented to video streaming over the Internet and broadcast TV. However, this standard did not achieve a great success.

ITU-VCEG and ISO-MPEG worked together again for the development of the ITU-H.264 or MPEG-4 Part 10 standard [ITU-T, 2003], usually called H.264/AVC, that was finally published in 2003. In H.264/AVC a great variety of coding tools were defined to achieve higher compression ratios and quality than previous standards. In doing so, it was able to cope with a lot of different applications, such as Internet streaming, broadcast TV, storage, etc. The Blu-Ray discs and several famous streaming applications over the Internet adopted this standard, increasing its popularity.

Continuing this line of work, both standardization organizations jointly released

in 2013 the most recent video coding standard, called H.265 or HEVC [ISO, 2013]. This standard was able to achieve a 50% reduction of the bit rate for the same quality compared with H.264/AVC [Vanne et al., 2012]. Several new tools were designed in HEVC aiming to support the most recent applications, as well as video resolutions above HD and the 3D video coding.

3.3 Overview of the H.264/AVC standard

The H.264/AVC standard [ITU-T, 2003] follows the hybrid DPCM/DCT block-based model and includes new features with which is able to reduce by half the bit rate generated by previous standards.

It should be noted that, similar to previous standards, H.264/AVC only specifies the syntax of an encoded video bit-stream and the decoding process. There is no a specification on the particular design of the video encoder and decoder. This helps the interoperability of the video coding systems based on H.264/AVC since to generate a standard-compliant bit-stream the encoder does not require to implement all the defined tools. However, the majority of the implementations of the H.264/AVC coding system include some basic functional elements, e.g., prediction, transform, and quantization. This flexibility allows each application to adapt the coding system configuration to its specific requirements, e.g., processor, memory, or network capabilities. Obviously, depending on the number of coding tools used, the coding process will find a different balance between the resulting coding efficiency and the complexity of the system.

3.3.1 H.264/AVC block diagram

The most typical H.264/AVC block diagrams are shown in Figures 3.1 and 3.2. The first one shows the encoder and the second the decoder.

As it can be seen, these block diagrams are very similar to those presented in Chapter 2. Now, the two prediction types (Intra and Inter prediction) are explicitly included in the diagrams while the remaining blocks are the same as those shown

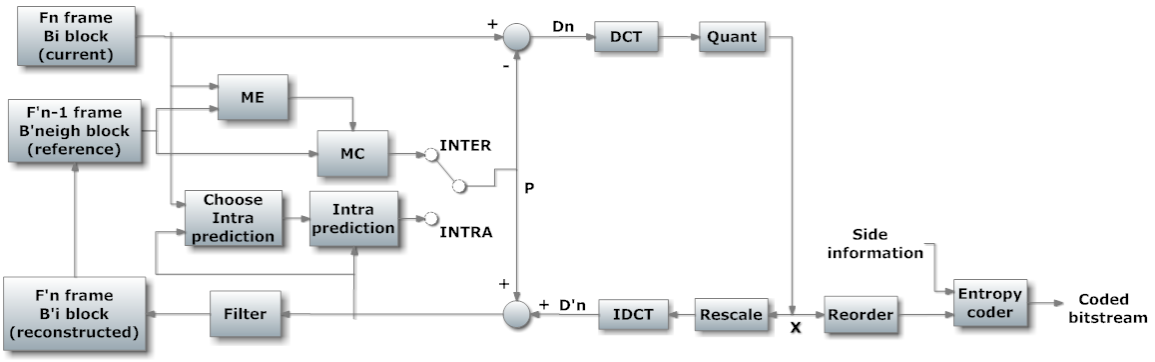


Figure 3.1: H.264 video encoder block diagram.

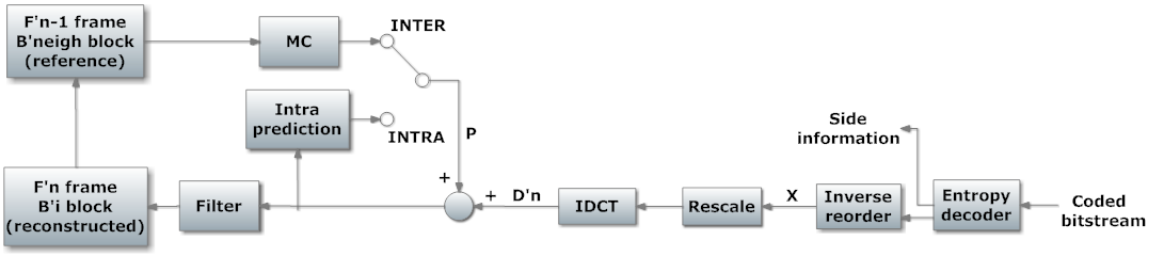


Figure 3.2: H.264 video decoder block diagram.

in Figures 2.4 and 2.5. We can find an additional filter block in the decoder and the reconstruction path of the encoder. This is usually called *deblocking filter* and it is responsible for reducing the blocking distortion in the reconstructed or decoded blocks.

3.3.2 Video format

The width and height of the luminance frames in the input of an H.264/AVC encoder must be a multiple of 16 and the chroma frame size must be multiple of 8 or 16, depending of the color sampling format considered (the interested reader is referred to [Richardson, 2003] for more details about color formats). This is due to the fact that the work unit in H.264/ACV is 16×16 pixels for the luminance component along with the associated pixels for the chroma components. Thus, for coding purposes the frame to be encoded will be divided into 16×16 pixel blocks, called *macroblocks* (MBs).

The encoder organizes the MBs in *slices*, each one containing an integer number of MBs that must be coded in raster order. Moreover, each frame could contain one or more slices that will be coded independently of the others, limiting in this way the error propagation. In this Thesis we use slices of the same size as the frames; thus, hereafter we will refer only to frames.

There exist several types of slices, as those types of frames described in Chapter 2 (Section 2.3.4.3); in fact, the I, P, and B types are all available in H.264/AVC. Moreover, there are two additional types of slices, called SP and SI. These two last are adequate in applications where transmission losses happen.

Figure 3.3 shows the data organization inside a slice. We find a slice header that indicates if the slice is I, P, B, SP, or SI. We also find information related to the coding process of every MB, e.g., the MB type (I, P, or B), the data required to build the prediction for that MB, and the coded residue.

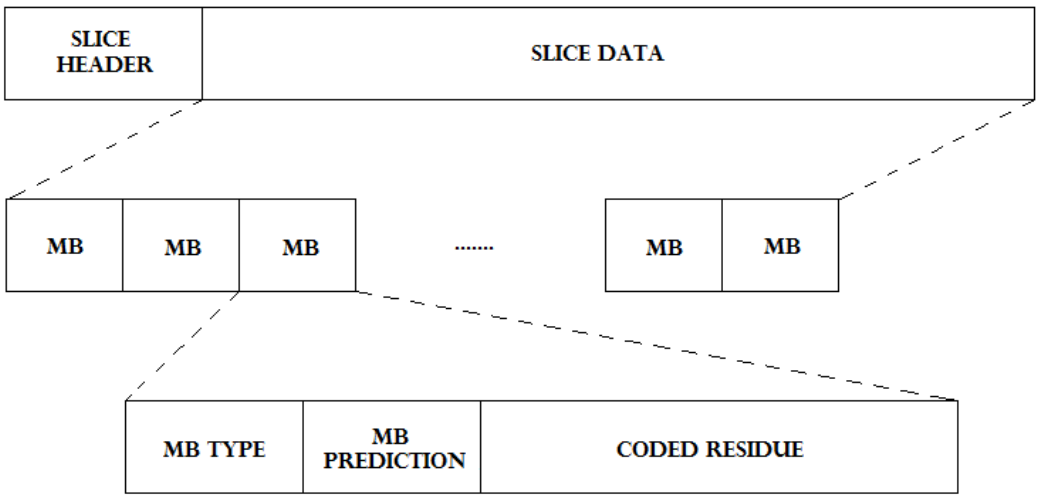


Figure 3.3: Slice data organization.

3.3.3 Intra prediction

The Intra prediction is obtained from a combination of the previously coded and reconstructed pixels in the neighborhood of the current MB (those located in the left, top, and upper left positions).

The Intra prediction can be calculated for the complete 16×16 pixel MB, or it can be further splitted down into 4×4 pixel blocks; both options are called *Intra modes*. In the first mode, four Intra prediction types are defined to form the prediction for the entire 16×16 pixel MB. In the second mode, there exist nine prediction types to form the prediction for every 4×4 block. These prediction types are different ways to combine the surrounding pixel values. Figures 3.4 and 3.5 illustrate these types for the Intra 16×16 and Intra 4×4 modes, respectively.

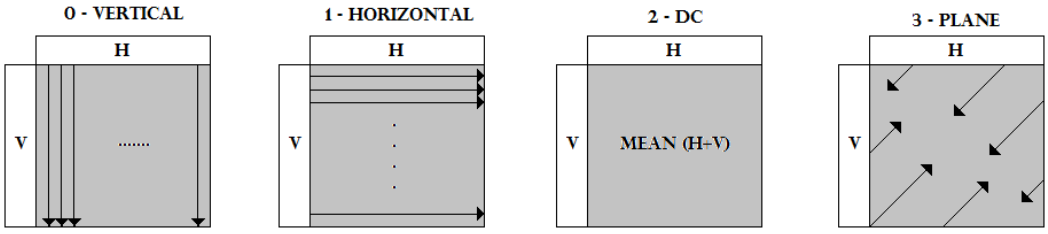


Figure 3.4: Intra 16×16 prediction modes.

In general, as it can be seen, most of the prediction types are based on directional predictions. Each direction is suitable to build a prediction in a block that presents texture in that direction.

Commonly, the Intra 4×4 mode is more useful in areas with high detail content or complicated textures. On the other hand, the Intra 16×16 mode is appropriate for homogeneous areas or those showing smooth variations.

3.3.4 Inter prediction

The Inter prediction is calculated from previously coded and reconstructed pixels in different frames. It should be noted that several tools included in this prediction

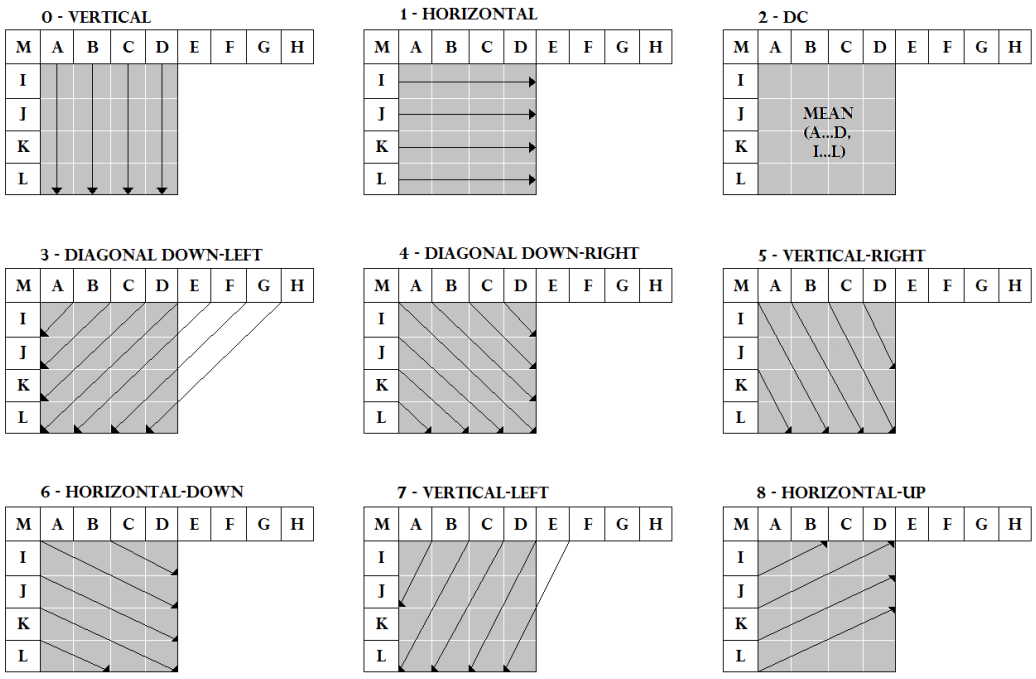


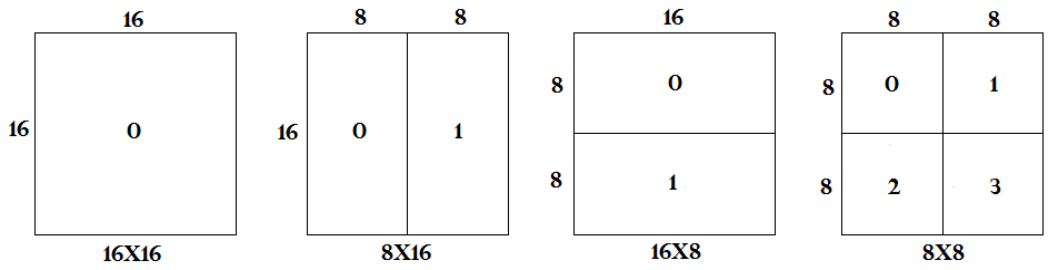
Figure 3.5: Intra 4×4 prediction modes.

turn out to be key differences with respect to the previous standard, e.g., variable block sizes to carry out the ME and MC processes, or quarter-sample resolution to obtain the prediction. All these tools are responsible for the improved efficiency of H.264/AVC. Next, these tools are described in detail.

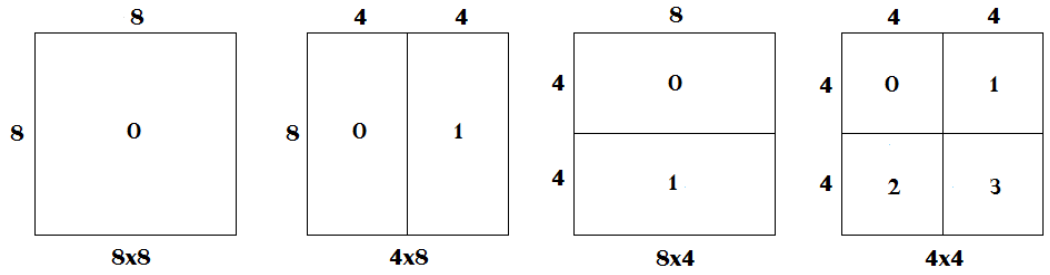
3.3.4.1 Inter prediction modes

To carry out the Inter prediction, several partition sizes for the MB can be used, called *Inter modes*. Specifically, the MB can be considered as a whole, or can be divided into two 16×8, two 8×16, or four 8×8 pixel partitions, as it can be seen in the upper part of the Figure 3.6. Additionally, when the 8×8 partition is evaluated, it can be further splitted, as it can be seen in the lower part of the Figure 3.6, i.e., the block can be considered as a whole of 8×8 pixel block, or it can be partitioned in 8×4, 4×8, or 4×4 pixel blocks. These modes are called the sub-macroblock (sub-MB)

partitions. The method of partitioning an MB into several blocks and sub-blocks of different sizes is called *tree-structured motion estimation*.



(a) Inter MB modes: 16×16, 8×16, 16×8, and 8×8.



(b) Inter sub-MB modes: 8×8, 4×8, 8×4, and 4×4.

Figure 3.6: Inter modes.

The Inter prediction is assessed for every partition of the corresponding Inter mode, e.g., if the 16×8 Inter mode is considered, the ME and MC processes are carried out in both the upper and lower 16×8 resulting partitions. Therefore, a different MV is obtained for each partition.

Thanks to this flexibility in the Inter prediction process, a high accuracy to represent the moving content is achieved, but at the expense of a very high complexity since many different alternatives must be evaluated by the encoder. Regarding the bit rate associated with every Inter mode, the large modes (Figure 3.6 (a)) require less MV information (as they are partitioned in few blocks and thus less number of MVs must be sent to the decoder), while the small modes (Figure 3.6 (b)) must send more information related to the MVs. On other hand, considering the accuracy of

the prediction, large modes lead to less accuracy representing the details in the movement of the objects, while small modes are more accurate. Reaching an appropriate balance for the representation of each MB, which is the goal of the encoder, is not an easy task.

To decide the best representation of the MB taking into account all the Intra and Inter modes, the encoder needs to evaluate all the Intra and Inter predictions available and select the optimal one. The encoder normally selects the best representation of the MB through a *Rate-Distortion Optimization* (RDO) method, that will be explained in Section 3.5.

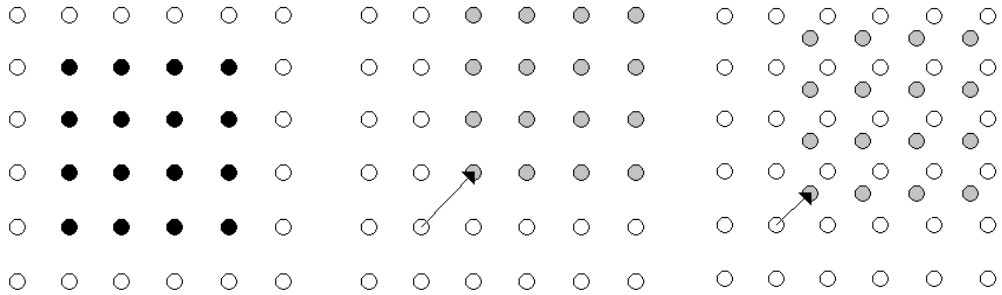
3.3.4.2 Motion vector

The first step of the ME process is to calculate the MV for every considered MB partition. The encoder also uses in this process the RDO method to seek, for each partition, the optimal pixel block in previously coded frames. The position of the selected block will be represented by means of the MV, which contains the relative coordinates of the best matching with respect to the position of the current block.

Usually, the area where the encoder searches for the matching block is restricted around the position of the current MB, avoiding the search in the complete frame. Moreover, the search is performed in all the reference frames available; however, typically, the number of reference frames is limited to avoid searching over frames very distant in time.

One of the main features of the ME process in H.264/AVC is that the MV can reach up to quarter-pixel resolution. The in-between pixel values have to be estimated by the encoder. In Figure 3.7 an example of integer (when the MV points at an actual pixel value) and fractional (when the MV points at a half or quarter-pixel position) MEs are shown.

To obtain these values the encoder uses interpolation from the actual neighboring pixels. First, the half-pixel positions must be calculated. To this end, the interpolated values are calculated from the adjacent integer samples by means of a six-tap FIR (Finite Impulse Response) filter with different weights depending on how far the



(a) Original 4×4 block. (b) Predicted 4×4 block with integer ME. (c) Predicted 4×4 block with fractional ME.

Figure 3.7: Example of integral and fractional predictions.

sample is from the position to be calculated. Then, the quarter-pixel values can be calculated by means of linear interpolation between the two adjacent half-pixel values.

With all these values calculated, the ME process can be carried out with very high accuracy, obtaining precise representations of the movement of the objects in the scene. This higher resolution notably improves the performance of previous standards, which only allowed half-pixel resolution. However, all these operations to obtain the fractional pixel values and to search through them also implies a notable increment of the computational complexity.

3.3.4.3 Motion vector prediction

The information concerning the MVs becomes a large proportion of the final bit rate generated by the encoder. To reduce it, the encoder takes advantage of the high correlation among MVs of neighboring blocks. Since nearby areas are likely to present similar movement, the redundancy among their MVs could be high and the encoder can profit from it to create an accurate prediction for each MV. Then, the difference between the prediction and the actual MV is calculated. The resulting differential MV will have lower magnitude than the original MV, and, consequently, will require fewer bits. This predicted MV (MVp) is calculated from MVs of previously coded neighboring blocks. In general, the median vector of the neighboring MVs is used as

MVp.

The decoder, using the same neighboring MVs as the encoder, will build the same MVp and, adding the differential MV that it is included in the bit-stream, will obtain the resulting MV.

3.3.4.4 Skip prediction mode

The Skip mode is a particular type of prediction available in P-frames for which no information is sent to the decoder, only a flag to signal that this MB is coded as Skip is included in the bit-stream.

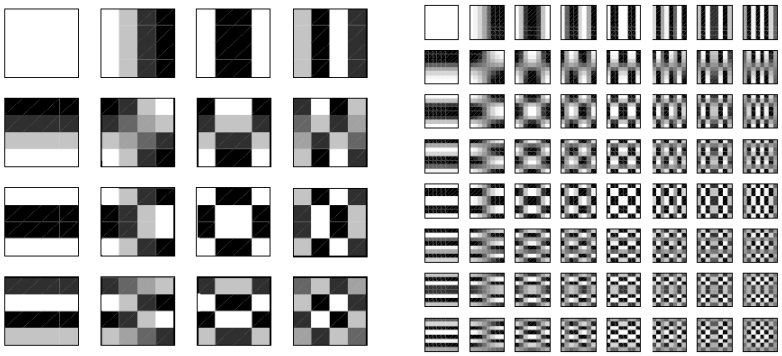
Specifically, the Skip mode has the following properties: (i) only one MV is considered, i.e., that of the 16×16 pixel partition size; (ii) the MV is the same as the MVp, so there is no information in the bit-stream related to the differential MV; (iii) the reference frame is considered the previous frame in display order; and (iv), all the DCT coefficients of the residue are zero. In doing so, the decoder just needs to copy the MB pointed by the MVp in the previous frame, and this will be the decoded MB.

This type of prediction is very suitable for areas without movement or, at least, low and similar movement as the neighboring areas. This coding tool is quite powerful since it significantly reduces the amount of bits necessary to represent MBs in quite common situations, leading to high coding efficiency.

3.3.5 Transform

Once the prediction is calculated following some of the previously explained procedures, it is subtracted from the MB to be encoded, obtaining a residual MB. This residual MB goes through a transformation process carried out with the DCT transform. The goal of the transform is to reduce the correlation in the residue data to obtain a more compact representation.

The output of the DCT will be a matrix of the same size as the input residue matrix but, now, each coefficient does not represent a residue pixel value, but the energy associated with a *basis function* of certain spatial frequencies. These *basis functions* are represented in Figure 3.8 for the 4×4 and 8×8 DCT transform.



(a) 4×4 DCT. (b) 8×8 DCT.

Figure 3.8: DCT *basis functions*.

Normally, to represent a residue block, just a few frequencies are required, and it is very common that higher frequencies have lower energies and vice-versa. Therefore, just a few non-zero coefficients will be obtained after quantization and, likely, around the low frequencies. In this way, the information is compacted and the bit rate resulting from encoding the transformed coefficients will be low.

H.264/AVC allows 4×4 and 8×8 DCT transforms, although the most common is the 4×4 size. Since this transform must be calculated for every possible representation of the MB (every block partition in all the available Inter and Intra prediction modes, reference frames, etc.), this process could result in a very high computational burden. To reduce it, H.264/AVC uses an approximation of the DCT based on additions and shifts, avoiding to use fractional operations and obtaining a very efficient implementation of the transform.

3.3.6 Quantization

The quantizer used in the H.264/AVC standard is very similar to an uniform quantizer, as that shown in Figure 2.7. Mathematically, the quantizer is defined as follows:

$$X_{ij} = round\left(Z_{ij}/Q_{step}\right), \tag{3.1}$$

where Z_{ij} is the actual DCT coefficient in the position ij of the transformed residue matrix, Q_{step} is the quantization step, and X_{ij} is the output of the quantizer, i.e., the quantized DCT coefficient, in the same position ij.

The Q_{step} is a key parameter of the quantizer. The number of possible output values changes depending on this parameter. As the Q_{step} grows, the amount of possible output values is lower and the errors between the input and the output of the quantizer will be larger, as it can be inferred from (3.1). Therefore, the coding losses grow with Q_{step}, while the produced bit rate is lower, as fewer codewords are required to represent the outputs. It should be noted that this balance is one key issue in video coding; in fact, there is no a closed solution for the optimal balance since it depends on the application.

In H.264/AVC, the parameter that the user can control to obtain different performances of the quantizer is the *Quantization Parameter (QP)*. This QP value is not the same as Q_{step}, but there is a univocal relationship between both parameters; thus, controlling the QP we also control the Q_{step}. Some of the specific values that can be taken by both parameters are shown in Table 3.1.

Q_{step}	0.62	1.25	2.5	5	10	20	40	80	160	240
QP	0	6	12	18	24	30	36	42	48	51

Table 3.1: Relationship between Q_{step} and QP.

The rescaling process (the inverse of the quantization) that must be carried out in the reconstruction path of the encoder (and also in the decoder) is simply defined as:

$$Y_{ij} = Z_{ij} \times Q_{step}, \tag{3.2}$$

where Y_{ij} is the reconstructed DCT coefficient in the position ij. The reconstructed Y values are different from the original X DCT coefficients since, as it can be seen in (3.1), there is a rounding operation that can not be reversed.

After the quantization process, the reordering of the quantized DCT coefficients is carried out by the encoder. The scanning method in H.264/AVC follows the pattern shown in Figure 2.8, except that this reordering is carried out for 4×4 pixel blocks.

3.3.7 Entropy coding

There are two types of entropy coders in H.264/AVC: the first is based on Context-based Adaptive Variable Length Coding (CAVLC) and the second is based on Context-based Adaptive Binary Arithmetic Coding (CABAC).

1. *Context-based Adaptive Variable Length Coding (CAVLC)*

   Some typical features of the quantized DCT coefficients after the reordering are exploited by this method. CAVLC performs a *run-level* coding to compact the zero coefficients that appear in the high frequencies of the block. Usually, the number of non-zero coefficients in neighboring blocks is highly related and this is exploited, by means of look-up tables, to encode them. Moreover, DC coefficients of neighboring blocks tends to be quite similar. Thus, the entropy coding of the DC coefficient is also carried out by means of look-up tables that will be selected taking into account the previously coded DC coefficients.

2. *Context-based Adaptive Binary Arithmetic Coding (CABAC)*

   This method divides the data to be encoded into binary syntax elements. For each one, a probability model (called *context*) is selected and adapted by means of local statistics. First, each element to be coded must be binarized. Then, the encoder selects the context model depending on the statistics of the recently coded elements. This context model indicates the probability of a bit being 0 or 1. Finally, the bit-stream for that element is generated using arithmetic coding and the context is updated. Thanks to these tools, the CABAC entropy coding method achieves higher compression ratios than CAVLC.

Following one of these two procedures, the encoder obtains the bit-stream representing the video sequence.

3.4 Overview of the HEVC standard

The latest video coding standard HEVC [ISO, 2013] also follows the block-based hybrid coding paradigm. As in H.264/AVC, in HEVC only the syntax of the bit-

stream is standardized, so that developers have total freedom when designing their standard coders for specific applications. However, there are some basic coding tools that are implemented in the majority of HEVC coders. In this Section we provide a brief explanation of these fundamental tools, focusing on those that are novel relative to the H.264/AVC standard and relevant for the development of our work.

3.4.1 HEVC block diagram

In Figure 3.9 we show a typical block diagram of the HEVC encoder.

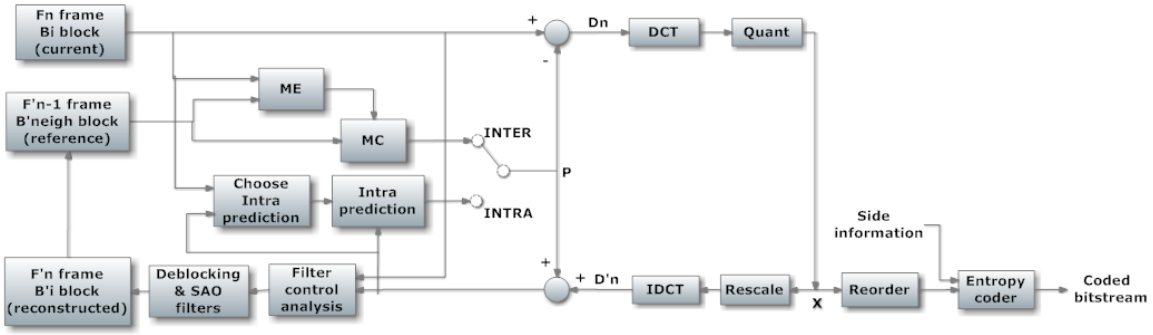


Figure 3.9: HEVC video encoder block diagram.

As it can be seen, most of the blocks are the same as those illustrated in the diagram of the H.264/AVC encoder. The Intra and Inter prediction blocks, along with the transformation and quantization blocks, are again the main components of the coding system. However, there are important differences in the performance and functionalities of these blocks compared with H.264, as it will be explained later.

Again, there is a deblocking filter in the reconstruction path of the encoder. Moreover, in HEVC, there is another filter called *Sample Adaptive Offset (SAO)* that consists of adding an offset value to the reconstructed samples based on look-up tables. The SAO filter aims to perform an additional refinement of the reconstructed samples in smooth and edge areas.

Moreover, in HEVC there are several tools to facilitate parallel processing. HEVC defines *tiles*, i.e., partitions of a frame into independent rectangular regions, *Wavefront Parallel Processing (WPP)*, which allows scanning the image using one thread

per row while maintaining CTBs dependencies, and *dependent slice segments*, which allow that associated data can be transported in different transport units.

3.4.2 Quadtree coding structure

In HEVC the work unit is called *Coding Tree Block (CTB)* and its size is selected by the encoder as 16×16, 32×32, or 64×64 pixels. The CTBs can be processed as a whole or can be further splitted into smaller blocks; whatever its size, the resulting block is called *coding unit (CU)*. The splitting is carried out by means of tree structures, called *quadtree coding structure*. The encoder selects the more suitable CU size depending on the features of the block through a RDO method. The dimensions of the CU can vary from 8×8 pixels up to the CTB dimensions, depending on the tree depth at which the CU is located. The higher depth, the lower the CU dimensions. Therefore, a CTB can consist of either only one CU or multiple CUs. The quadtree structure is illustrated in Figure 3.10.

The prediction for each CU can rely on different partition sizes, called *prediction units (PUs)*, being Intra or Inter depending on whether the CU uses spatial or temporal prediction, respectively.

Regarding the transformation stage, the CU can be considered as the root of another quadtree where the transformation of the residue is carried out. The size of the *transform units (TUs)* can vary from 4×4 up to the CU size, depending on the depth level at which the TU is located. In contrast to H.264/AVC, HEVC allows that a TU spans across multiple PUs for Inter prediction to maximize the coding efficiency.

3.4.3 Intra prediction

There are two possibilities to divide the CU into PUs: first, when the CU size is larger than the minimum, the PU is always equal to the CU size; second, when the CU size is the minimum, the PU can be the same size as the CU or it can be splitted into four equal-size blocks. Following the typical HEVC notation, the PU size is denoted

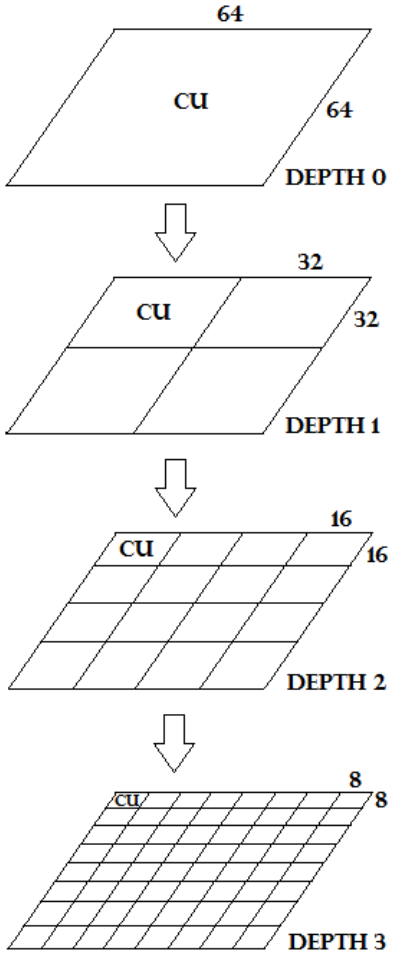


Figure 3.10: Quadtree coding structure of a CTB: from one 64×64 CU to 64 8×8 CUs.

as 2N×2N when the PU is not splitted and N×N otherwise.

The Intra prediction in HEVC has been significantly enhanced, increasing the number of prediction directions. All the possible directions are shown in Figure 3.11. Moreover, the planar and the DC modes are also available.

The angles defined for Intra prediction in HEVC are designed to provide a dense cover in all the possible orientations, improving the accuracy of this kind of prediction.

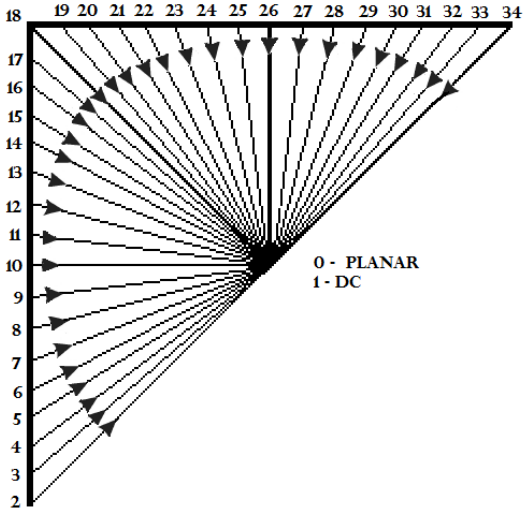


Figure 3.11: Orientations for Intra prediction in HEVC.

3.4.4 Inter prediction

3.4.4.1 PU partitioning

For Inter prediction, each CU can be divided into several PUs. Figure 3.12 shows all the possible partitions.

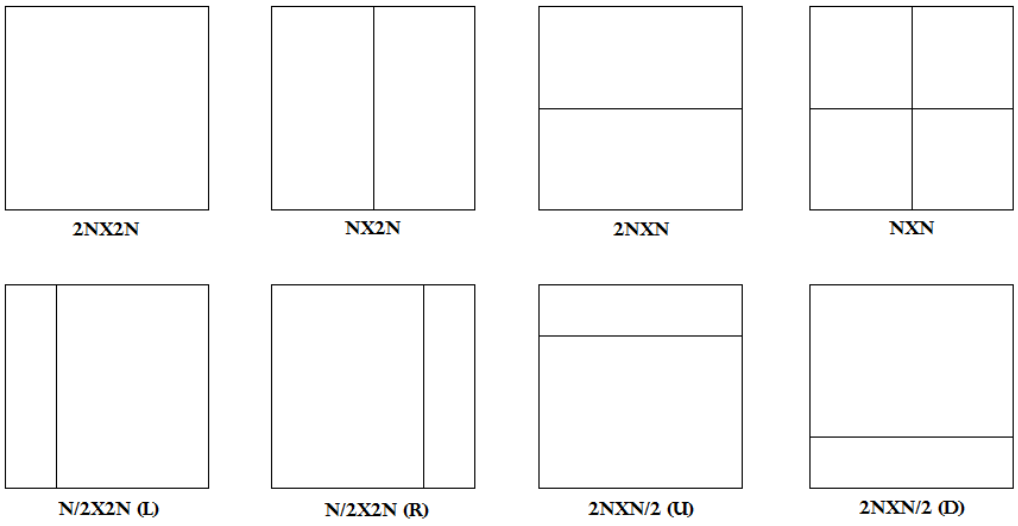


Figure 3.12: PU partitions for Inter prediction mode.

The Inter mode 2N×2N indicates the case in which the CU is considered as a whole, while the N×2N and 2N×N modes refers to two equal-size vertical and horizontal partitions. The N×N mode is only supported when the CU depth is the maximum (i.e., the CU size is minimum), and it refers to a four equal-size square division. Moreover, there are some asymmetric partitions available, denoted as N/2×2N (L), N/2×2N (R), 2N×N/2 (U), and 2N×N/2 (D) modes (where L, R, U, and D stand for left, right, up, and down partition), which provide an increased versatility for the encoder to accurately represent the movement. The asymmetric modes are only available when N is larger or equal to 8. As it can be seen, in any asymmetric PU, one partition has the height or width N/2, while the other partition has a height or width of 3N/2.

In HEVC the available CU sizes and partitions are many more than those available in H.264/AVC. In fact, for each CU size, the PU partitions shown in Figure 3.12 are used to represent more suitably the movement of such CU; thus, this tool enhances the accuracy of the ME process, but at the expense of a high increase of the computational complexity with respect to H.264/AVC.

The HEVC encoder must select the best coding option among all the available coding modes, and this is carried out through a RDO process.

3.4.4.2 Merge prediction mode

Similar to the Skip mode defined in H.264/AVC, HEVC includes a Merge prediction mode to take advantage of the motion information from neighboring blocks, trying to save bit rate in the coding process. However, there are several differences between Merge and Skip modes.

The Merge mode needs to identify a reference frame and an MV candidate, so it needs to transmit some index information, while the Skip mode assumed some predefined values. In this mode there is a so called *MV competition scheme*, where the best MV is selected among several candidates. The set of possible candidates consists of MVs coming from spatial and temporal neighbors, and other generated candidates. Figure 3.13 shows the positions of the five spatial candidates, evaluated

in the illustrated order.

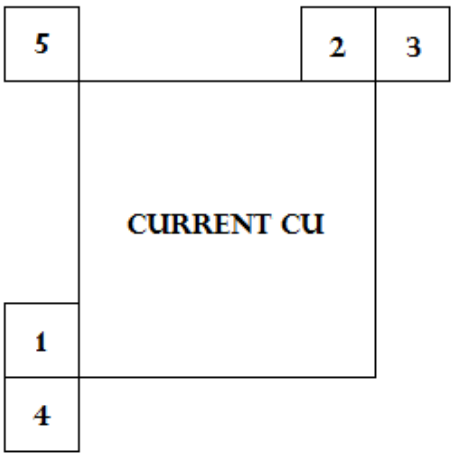


Figure 3.13: Positions of spatial candidates in the Merge mode.

For the MV temporal candidate, the right bottom position outside the collocated PU in the reference frame is used if available. If not, the PU in the center position is used. Furthermore, HEVC gives more flexibility to choose the temporal candidate since allows for selecting the reference frame index.

The maximum number of candidates C is specified in the slice header. If the number of spatial and temporal candidates is larger than the maximum, the temporal candidate and $C - 1$ spatial candidates are considered, whereas if the number of candidates is less than C, additional candidates are generated. The way to generate additional candidates depends on the type of frame considered. For P frames, for example, MVs with zero displacement associated with different reference indexes are added to the candidate list. In HEVC, the Skip mode is considered a special type of Merge mode.

3.4.4.3 Fractional sample interpolation

HEVC also supports MVs with up to quarter-pel accuracy. However, this process has been improved compared to that of the H.264/AVC, avoiding rounding operations

when generating the fractional locations, obtaining higher accuracy and simplifying the interpolation step.

The fractional sample interpolation for luminance pixels uses an eight-tap filter to calculate the half-pel positions and a seven-tap filter for the quarter-pel positions.

Figure 3.14 illustrates the fractional pixel locations. The samples $A_{i,j}$ represent the luminance values in integer positions, the remaining samples refer to half and quarter-pel positions. The samples $a_{0,j}$, $b_{0,j}$, $c_{0,j}$, $d_{i,0}$, $h_{i,0}$, and $n_{i,0}$ are obtained from the $A_{i,j}$ values with the corresponding eight or seven-tap filter, depending on whether the corresponding sample is located in a half or quarter-pel position, respectively. The samples $e_{0,j}$, $f_{0,j}$, $g_{0,j}$, $i_{0,j}$, $j_{0,j}$, $k_{0,j}$, $p_{0,j}$, $q_{0,j}$, and $r_{0,j}$ are obtained from the adjacent values with the corresponding filters.

3.4.4.4 Motion vector prediction

The HEVC encoder also codes the difference between the MV found in the ME process and a MVp (a predicted MV). Similar to the Merge mode, the encoder can select the MVp from multiple candidates, thus an index to identify the selected candidate must be also included in the bit-stream.

Only two spatial candidates are selected among the five candidates represented in Figure 3.13. The first candidate is selected from the left neighbors (the blocks numbered as 4 and 1 in the figure), and the second from the upper (the blocks 3, 2, and 5 in the figure), depending on their availability. If several neighbors are available, the candidate is selected in the order indicated by the labels.

When the number of candidates is lower than two, the temporal MV candidate is included, or even a zero MV, if necessary.

3.4.5 Transformation and quantization

Once the predicted CU is obtained, the difference between the original and predicted blocks is calculated, obtaining the residue. The residual CU can be considered as the root of another quadtree structure where the transformation process is carried out. In this way, the residue can be divided in multiple TUs, from the CU size down to 4×4

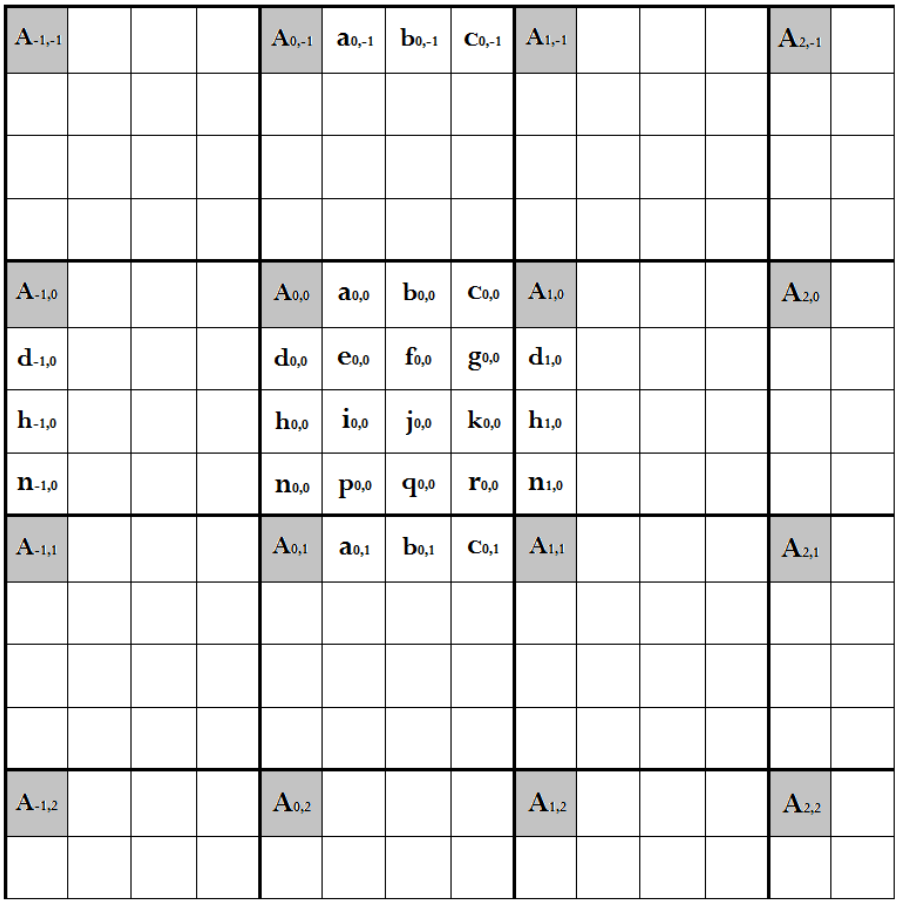


Figure 3.14: Fractional sample positions for interpolation.

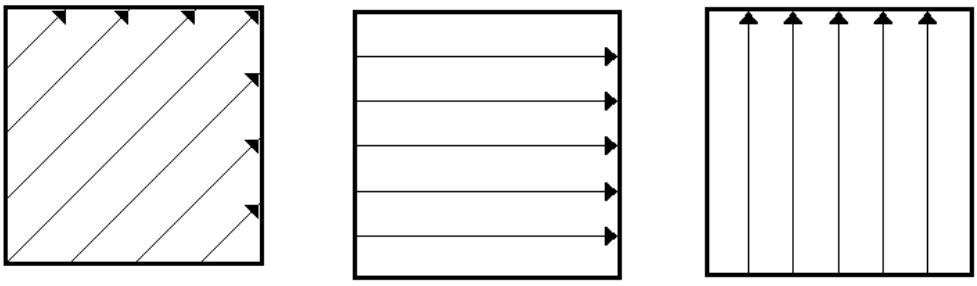
(it should be noted that the maximum size for a TU is 32×32; thus, if the selected CU size is 64×64, the CU will be divided into 4 TUs to carry out the transformation).

The two-dimensional transform is calculated from two one-dimensional transforms in the horizontal and vertical directions. The elements of the core transform matrix are obtained by approximating scaled DCT basis functions, limiting their dynamic range for computation purposes and simplifying the mathematical operations, while the accuracy is maximized. The elements of the matrices of each transform size can be derived from the 32×32 matrix, which is the only one specified in the standard.

Regarding the quantization step, HEVC uses the same scheme controlled by QP

as in H.264/AVC. The range of the QP values is still from 0 to 51 and its relationship with Q_{step} is the same shown in Table 3.1.

For the scanning of the quantified coefficients, HEVC uses an adaptive system to select the most adequate scanning pattern for every specific situation. The coefficient scanning is carried out always in 4×4 blocks, independently of the optimal TU size. In Intra prediction with TU sizes of 4×4 or 8×8, the HEVC encoder can select the scanning method among the three possibilities shown in Figure 3.15. As it can be seen, they are directional scannings: diagonal up-right, horizontal, and vertical. The selection of the scanning method depends on the optimal direction selected for the Intra prediction. Specifically, the vertical scan is used when the optimal Intra direction is near to horizontal, the horizontal scan is selected with directions near to vertical, and the diagonal up-right scan is used for the remaining Intra directions.



(a) Diagonal up-right. (b) Horizontal. (c) Vertical.

Figure 3.15: Coefficient scanning methods in HEVC.

In Intra prediction with TU sizes of 16×16 or 32×32 and in the Inter prediction with any TU size, the 4×4 diagonal up-right scanning method is used for all sub-blocks.

3.4.6 Entropy coding

HEVC only defines CABAC as entropy coding method. The main steps for implementing CABAC remain unaltered with respect to H.264/AVC; however, the differences in the coding process between both standards are exploited to improve the

performance of the CABAC method.

For example, the depth of the CU and TU quadtress are used to derive the context model of various syntax elements. The number of contexts used in HEVC is lower than in H.264/AVC; however, the CABAC design in HEVC provides better compression, as dependencies between the coded data are exploited to additionally improve its performance.

Regarding the coding of the residue coefficients, HEVC codes the position of the last non-zero coefficient, a significance map (that indicates which transform coefficients have non-zero values and their positions), sign bits, and levels for each non-zero coefficient. Moreover, several changes have been made to manage the increased size of the transform, exploiting the redundancy between adjacent 4×4 sub-blocks.

3.5 Rate-Distortion Optimization

The H.264/AVC and HEVC encoders need to find the optimal coding options among all the available (QP, prediction mode, block size, MV, reference frame, etc.). Specifically, the encoder will select the coding options that minimize the distortion subject to a certain rate constraint. This means that the selected coding options must be those that generate an amount of bits lower or equal than the rate constraint at the same time that minimize the distortion.

To carry out this selection process, the implementations of the H.264/AVC and HEVC standards usually use a RDO method. This process can be formulated as follows:

$$\min_{\boldsymbol{\theta}} \{D(\boldsymbol{\theta})\} \text{ subject to } R(\boldsymbol{\theta}) \leq R_c, \tag{3.3}$$

where $\boldsymbol{\theta}$ is a combination of coding parameters (block size, prediction mode, MV, reference frame, QP, etc.); $D(\boldsymbol{\theta})$ represents the distortion associated to this $\boldsymbol{\theta}$ combination; $R(\boldsymbol{\theta})$ is the amount of bits generated by the encoder with $\boldsymbol{\theta}$, including residual transform coefficients, side information, and headers; and, finally, R_c is the rate constraint.

Using Lagrange formulation, the constrained optimization problem stated in (3.3)

can be expressed as an unconstrained optimization problem [Everett, 1963]. To that purpose, a R-D cost function J is built that weights the distortion and the rate terms by means of a Lagrange multiplier λ. The goal is to find the coding options $\boldsymbol{\theta}$ that minimize this J cost:

$$\min_{\boldsymbol{\theta}} \{J\}, \text{where } J(\boldsymbol{\theta}) = D(\boldsymbol{\theta}) + \lambda R(\boldsymbol{\theta}), \tag{3.4}$$

where the Lagrange multiplier λ balances the relative importance of distortion and rate obtained with the coding options $\boldsymbol{\theta}$. Solving (3.4) for a particular λ value, we obtain an optimal solution $\boldsymbol{\theta}^*(\lambda)$ of the original RDO problem (3.3) for a particular rate constraint $R_c = R(\boldsymbol{\theta}^*)$.

This solution involves to evaluate all the possible combinations for $\boldsymbol{\theta}$ in each block (MB or CTB, depending on the considered standard) to choose the optimal one, but this is not feasible due to the incredibly high computational cost that this would suppose (note that it would be necessary to evaluate the final distortion and rate for each block and each combination of coding parameters). In practical implementations of the H.264/AVC and HEVC standards, several simplifications are made in order to obtain a lighter RDO process.

One of the simplifications is to consider the decisions in each block independent. Although this hypothesis is not true because the $\boldsymbol{\theta}^*$ selected for a block actually depends on those of the previously coded blocks, this simplification is necessary to obtain a practical solution of the RDO process. Moreover, decisions at different coding stages (e.g., prediction mode, MV, etc.) are also considered independent.

Furthermore, the practical implementations of the optimization process divide the RDO process in two steps. The first is devoted to the selection of the optimal reference frame and MV and the second is responsible for selecting the best prediction mode.

The RDO process to select the best reference frame (Ref) and MV is mathematically defined as follows:

$$J_{motion} = SAD(MV, Ref) + \lambda_{motion} R_{motion}(MV, Ref), \tag{3.5}$$

where SAD is the sum of the absolute differences between the original and predicted blocks, which is the distortion measure used in this first step; λ_{motion} is the Lagrange multiplier; and R_{motion} is an estimation of the amount of bits required to code the MVs. As it can be seen, in (3.5) there are several simplifications: the distortion is calculated with the predicted block instead of the reconstructed one, and the rate is only an approximation of the actual rate. Taking into account that this formulation must be evaluated in every pixel location considered in the ME process (for all the Inter prediction modes, partitions in a block, available positions in the search range, and reference frames), this helps to alleviate the computational burden of this process.

Once the reference frames and MVs have been selected for each Inter prediction mode, the encoder seeks for the optimal prediction mode k among all the Inter and Intra modes, what is usually called *mode decision* (MD) process. In the H.264/AVC standard, this process means to find the optimal MB prediction and partition size, while in HEVC it refers to the optimal prediction and the complete arrangement in the quadtree coding structure, including CU, PU, and TU sizes. For this second step the encoder uses a different R-D cost function, defined as follows:

$$J_{mode,k} = \quad SSD(\{MV\}_k, \{Ref\}_k, k) + \lambda_{mode} R(\{MV\}_k, \{Ref\}_k, k), \qquad (3.6)$$

where SSD is the sum of the squared differences between the original and reconstructed blocks, which is the distortion measure used in this second step; λ_{mode} is the Lagrange multiplier (different from that used in (3.5)); and R is the rate to code the headers, MVs, indexes of the reference frames, and the residual coefficients. As it can be seen, this second step is more similar to the original formulation, as the distortion and rate measures are calculated without simplifications.

Both RDO methods are related by means of their Lagrange multipliers as follows [Sullivan and Wiegand, 1998]:

$$\lambda_{motion} = \sqrt{\lambda_{mode}}. \qquad (3.7)$$

Moreover, in order to avoid checking every QP value available in the encoder,

a relationship between the Lagrange multiplier λ_{mode} and QP was experimentally obtained [Sullivan and Wiegand, 1998]:

$$\lambda_{mode} = C \times 2^{\frac{QP}{3}}, \tag{3.8}$$

where C is a constant value that depends on the slice type and the encoder configuration. Thus, for a given QP value, the encoder can easily obtain both Lagrange multipliers λ_{mode} and λ_{motion}. Although other relationships between QP, λ_{mode} and λ_{motion} values have been investigated in the state-of-the-art (e.g., [de Suso et al., 2014, Li et al., 2009]), the formulations in (3.7) and (3.8) are usually applied in practical implementations of the standards.

As it can be seen, these simplifications allow the video coding system to obtain a near optimal representation of each block, $\hat{\boldsymbol{\theta}}^*$; however, the large amount of coding options still involves a high computational burden of the RDO process, resulting in the bottleneck in the video coding process.

Chapter 4

Complexity Control in H.264/AVC

In this chapter we face the complexity control problem in the H.264/AVC standard, proposing a novel method that relies on a hypothesis testing that can handle time-variant content and target complexities. Specifically, it is based on a binary hypothesis testing that decides, in an MB basis, whether to use a low- or a high-complexity coding model. Gaussian statistics are assumed so that the probability density functions (PDFs) involved in the hypothesis testing can be easily adapted. The decision threshold is also adapted according to the deviation between the actual and the target complexities. The proposed method was published in IEEE Transactions on Multimedia [Jimenez-Moreno et al., 2013].

This chapter is organized as follows. In Section 4.1 we present a brief introduction. Section 4.2 gives a review of the most relevant contributions to the complexity control problem in H.264/AVC. Section 4.3 explains in detail the proposed method. The experiments conducted to prove the strengths of the method and a discussion of the results are presented in Section 4.4. Finally, Section 4.5 summarizes our conclusions.

4.1 Introduction

The conception of algorithms capable of adapting their computational complexity (obviously in exchange for performance, memory, delay, etc.) to those supported by

specific devices becomes an important challenge that have received some attention and that will continue to be of interest in years to come.

Video coding is one of the numerous signal processing systems that, in some scenarios, are required to be complexity-adaptive. Although many research efforts have been devoted to reduce the complexity of video compression algorithms [Tourapis and Tourapis, 2003, Zhu et al., 2002, Zhang et al., 2003, Choi et al., 2003, Li et al., 2005, Gonzalez-Diaz and Diaz-de Maria, 2008, Grecos and Mingyuan, 2006, You et al., 2006, Kuo and Chan, 2006, Saha et al., 2007, Zhou et al., 2009, Martinez-Enriquez et al., 2007, Martinez-Enriquez et al., 2009, Martinez-Enriquez et al., 2010, Martinez-Enriquez et al., 2011, Lu and Martin, 2013, Kim et al., 2014, Yusuf et al., 2014], only a few works have been devoted to actually control the complexity [Ates and Altunbasak, 2008, Gao et al., 2010, Kannangara et al., 2008, Huijbers et al., 2011, Vanam et al., 2007, Vanam et al., 2009, Su et al., 2009, Tan et al., 2010, Kannangara et al., 2009, da Fonseca and de Queiroz, 2009, da Fonseca and de Queiroz, 2011, Li et al., 2014]. In this chapter, the problem of complexity control is tackled in the framework of the H.264/AVC standard.

We propose an algorithm capable of keeping the H.264/AVC encoder complexity around a certain externally provided target value with minimum losses in terms of coding efficiency, even when the target complexity is very low. The proposed approach has been devised to satisfy the following specifications: low miss-adjustment error with respect to the target complexity, capability to adapt to a time-variant complexity target and to the video content, and capability to operate on a large dynamic range of target complexities and to work with any image resolution.

4.2 Review of the state-of-the-art

Some of our previous works were focused on the complexity reduction problem [Martinez-Enriquez et al., 2009, Martinez-Enriquez et al., 2010, Martinez-Enriquez et al., 2011]. Despite not being able to achieve a target complexity, they establish a starting point for our research work on complexity control. First, in these works we studied

the design of several classifiers to reduce the encoder complexity, from decision thresholds to linear classifiers, always obtaining reasonably good results in terms of time saving and without incurring significant encoding efficiency losses. Moreover, several statistical analysis of the input features to these classifiers were carried out, helping us to understand the encoder behavior and to identify the more relevant variables in the MD problem. This previous work motivates us to design a complexity control algorithm.

Focusing now on the complexity control problem, the most common approach involves adding a complexity term to the cost functions that are minimized in the RDO process. In [Ates and Altunbasak, 2008], an estimation of the high frequency content of a block and a target complexity are included in a novel cost function, so that the ME process relies on it to decide which partitions are taken into account for each MB. In [Gao et al., 2010], modified versions of both J_{motion} and J_{mode} cost functions were proposed by adding a complexity term that is based on the computation time and the number of instructions required. Moreover, the modes are rearranged according to a texture analysis, so that, given an available complexity for an MB, the encoding process picks modes according to the resulting arrangement, and stops whenever the accumulated complexity exceeds the target complexity. Once a subset of modes has been selected in this manner, the modified cost functions are used to decide on the best representation for the MB. It is also worth mentioning that this method requires a costly *off-line* estimation of the Lagrange multipliers involved in the cost functions. In [Kannangara et al., 2008], an algorithm that relies on encoding-time statistics to reach a given complexity target was suggested. In particular, the algorithm estimates the encoding complexity from a buffer occupancy measurement and manages this complexity by means of a Lagrangian rate-distortion-complexity cost. Additionally, the encoder drops frames when the complexity target cannot be met. In [Huijbers et al., 2011] a complexity scalable video encoder that is capable of adapting *on the fly* to the available computational resources was presented. Specifically, this algorithm works at both frame and MB levels. At the frame level, the algorithm decides the maximum number of SAD calculations according to the com-

plexity budget. At the MB level, the complexity budget is allocated among the MBs in proportion to the distortion of the co-located MBs in previous frames. In [Vanam et al., 2007], an algorithm capable of finding an appropriate encoder configuration was described. Given a working bit rate, it finds optimal operating points taking into account distortion and complexity. The authors propose two fast approaches that do not require an exhaustive evaluation of encoder configurations. An extension of this work is presented in [Vanam et al., 2009] following the same principles. In [Su et al., 2009], an allocation of computational resources based on a virtual buffer was proposed. Additionally, to guarantee that the used resources do not exceed the estimated ones, two complexity control schemes are defined, one on the ME and the other on the MD processes. For the ME, a search path and a termination point are defined according to R-D considerations and the allocated complexity. For the MD, a search order and a termination point are defined according to the most frequent modes in neighboring MBs and the allocated complexity. In [Tan et al., 2010], the MBs in a frame are encoded using only Intra and Skip modes. Then, the encoding of the MBs producing the highest costs is further refined using additional modes. The number of mode decisions is controlled by means of a parameter that allows this method to be scaled for different complexity targets. In [Kannangara et al., 2009] the Bayesian decision theory was used for complexity control. In particular, a threshold to comply with an average target complexity level is determined using a probability model where the corresponding cumulative density functions are estimated based on motion measurements and the QP value. To this purpose, an *off-line* precomputed relationship among these parameters is required. This method is limited to Skip/non-Skip decisions.

The works described so far were tested on QCIF and CIF resolutions, since complexity control was considered attached to low-power devices, which were not able to work with higher resolutions. Nowadays, however, the fast growth in computational power has made even hand-held devices capable of working with higher resolutions. The works by Queiroz et al. ([da Fonseca and de Queiroz, 2009, da Fonseca and de Queiroz, 2011]) tackle the complexity problem for higher resolutions. In [da Fon-

seca and de Queiroz, 2009], the complexity is controlled by allowing only for a subset of modes in the MD process. Specifically, the most likely modes are sorted, and only those that do not exceed a pre-established complexity limit are evaluated. In [da Fonseca and de Queiroz, 2011] the values of distortion, rate, and complexity achieved by a set of specific encoder configurations are collected by means of an *off-line* training process. These values are tabulated and a desired level of complexity is reached by applying the corresponding encoder configuration. The weakness of this *off-line* training process is the difficulty of adapting the model to time varying conditions in both complexity requirements and video content.

The algorithm proposed in this Thesis, as a few of the previously mentioned ([Kannangara et al., 2008, Huijbers et al., 2011]), relies on a parameter estimation process that is carried out *on the fly*, avoiding both the generalization problems inherent to an *off-line* estimation and the computational cost associated with the training process. In this manner, the algorithm can easily adapt to changes in both target complexity and video content. As a result, the proposed method is simple and capable of efficiently operating on different video contents and resolutions and on changing complexity targets, exhibiting quite remarkable convergence properties. Furthermore, these high levels of simplicity and flexibility are achieved in exchange for acceptable losses in coding efficiency. Moreover, our proposal deals with resolutions from QCIF to HD.

4.3 Proposed method

4.3.1 Motivation and Overview

The proposed method to control the complexity of the H.264/AVC encoder is based on a hypothesis testing in which the decision threshold is automatically set to reach an externally-provided target complexity level. This approach has been adopted for two reasons: 1) the mathematical definition of the hypothesis testing allows for defining a cost policy adapted to the specific problem at hand, thus providing a valuable flexibility and adaptability; and 2) as we will show in the following sections,

this approach is effective to reduce the complexity while maintaining a high coding efficiency level.

We propose to make a binary decision in each MB, deciding between low- or high-complexity coding models; thus, the proposed algorithm relies on a binary hypothesis testing. These two proposed coding models are defined as follows:

1. In the low-complexity model, the MB can be encoded as Skip, Inter 16×16, or Intra 16×16. The reason to choose these three modes is the following: for the algorithm to meet tough complexity constraints, the amount of modes in the low-complexity level must be kept as low as possible. Therefore, it would have been desirable for this hypothesis to involve only the Skip mode, which does not require ME; however, considering only the Skip mode would have led to significant losses in coding efficiency. Consequently, to avoid these efficiency losses and still keep the complexity at reasonably low levels, the Inter 16×16 mode had to be included. Furthermore, the Intra 16×16 mode had to be included as well to achieve a satisfactory performance in those cases where the ME process does not work properly, i.e., when the penalty in coding efficiency for not allowing Intra modes is high.

2. In the high-complexity model, the MB can be encoded as any of the available Inter or Intra modes. In this case, as all the complexity resources are available to encode a MB, all the modes are evaluated to select the optimal among them.

Once all MBs in a frame have been encoded, the complexity control algorithm must check the achieved complexity and compute the deviation from the target. From this deviation, the complexity control algorithm adjusts the decision threshold of the hypothesis testing, so that this new threshold is used for the next frame to be encoded. The flowchart in Figure 4.1 summarizes the whole process.

Mathematically, the formulation of the hypothesis testing derives from the Bayesian decision theory, where the optimal decision given an observation x is the one that provides the lowest mean cost, which can be expressed as follows:

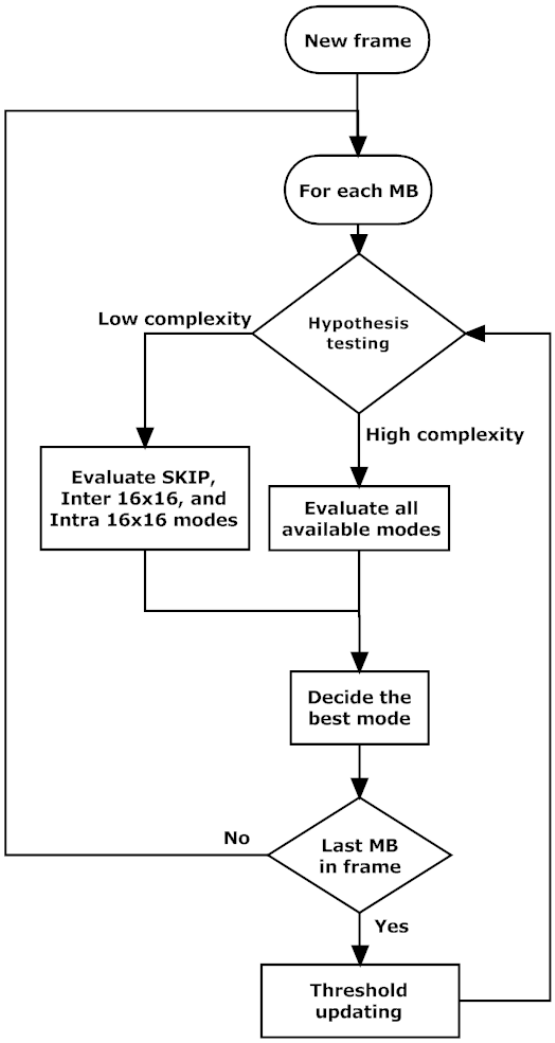


Figure 4.1: Flowchart of the proposed algorithm for complexity control in H.264.

$$D_{j^*} = \arg\min_j \sum_i C_{ji} \Pr_x(x|H_i) \Pr(H_i), \tag{4.1}$$

where D_{j^*} is the optimal decision, which is the one that minimizes a weighted sum over all possible hypothesis i of the costs C_{ji} of deciding j when the correct hypothesis is i. The weights in the sum are given by the likelihoods of obtaining the observation x given the hypothesis H_i ($\Pr_x(x|H_i)$) and the *a priori* probability of each hypothesis ($\Pr(H_i)$).

From this formulation, we can define a binary hypothesis testing with two possible hypothesis H_0 and H_1, and the two corresponding decisions D_0 and D_1. For this particular case, the mean costs associated with deciding either D_0 or D_1 are defined as follows:

$$\begin{aligned}\overline{C}_0(x) &= C_{00} \Pr_x(x|H_0) \Pr(H_0) + C_{01} \Pr_x(x|H_1) \Pr(H_1), \\ \overline{C}_1(x) &= C_{10} \Pr_x(x|H_0) \Pr(H_0) + C_{11} \Pr_x(x|H_1) \Pr(H_1).\end{aligned} \tag{4.2}$$

Finally, according to the previous expressions, the Likelihood Ratio Test (LRT) for this problem can be written as:

$$\frac{\Pr_x(x|H_1)}{\Pr_x(x|H_0)} \gtrless_{D_0}^{D_1} \frac{(C_{10} - C_{00})}{(C_{01} - C_{11})} \frac{\Pr(H_0)}{\Pr(H_1)}. \tag{4.3}$$

The following subsections explain in detail the main building blocks of the proposed method. First, we focus on a statistical analysis of the R-D costs to gain insight into the feasibility of the proposal and make the best possible design.

4.3.2 Statistical Analysis

Different features have been used in the literature to make an early mode decision. We proved in [Martinez-Enriquez et al., 2010] that the J_{mode} cost is one of the most informative features for this purpose. The analysis in this regard is presented next.

With the aim of studying the viability of using J_{mode} to design early stops, the conditional PDFs of J_{mode} for each particular mode k ($J_{mode,k}$), given that it is the optimal one (k^*), i.e., $\Pr_J(J_{mode,k}|k^* = k)$, has been analyzed. In addition, the *a priori* probability that the k^{th} mode is the optimal one, $\Pr(k^* = k)$, has been estimated. For this purpose, we have used the JM reference software[1] on a set of eight CIF and six QCIF video sequences encoded with different QP values (28, 32, 36 and 40) and an IP GOP structure.

[1] JVT H.264/AVC reference software v.10.2 [Online]. Available: http://iphome.hhi.de/suehring/tml/download/old_jm/.

	(A) Mean value				(B) Standard deviation				(C) *A priori* probability			
	QP 28	**QP 32**	**QP 36**	**QP 40**	**QP 28**	**QP 32**	**QP 36**	**QP 40**	**QP 28**	**QP 32**	**QP 36**	**QP 40**
Skip	5044	9493	17811	32648	2508	5348	11456	23262	0.600	0.655	0.709	0.766
16×16	7240	14030	27549	55647	4212	84095	16606	31991	0.107	0.102	0.105	0.107
16×8	7562	16032	32576	64743	4077	80403	15774	31705	0.043	0.044	0.043	0.039
8×16	7814	16602	34782	68942	4451	88141	17439	32442	0.043	0.044	0.044	0.039
P8×8	12780	25029	48633	91776	5427	10124	18441	32642	0.201	0.147	0.092	0.042
8×8	1290	2532	4938	9540	965	1959	4047	7998	0.825	0.865	0.904	0.941
8×4	2453	5046	10355	20007	1508	2986	5840	11255	0.067	0.056	0.041	0.028
4×8	2661	5600	11185	22465	1598	3107	5969	11655	0.078	0.062	0.047	0.028
4×4	3960	8195	16569	34219	1587	3015	5796	10103	0.029	0.016	0.006	0.001
Intra	1168	1866	3793	9537	435	315	827	2117	0.004	0.004	0.005	0.005

Table 4.1: Detailed R-D cost analysis for *Paris* (CIF).

	Foreman (CIF)	**Akiyo (CIF)**	**Container (CIF)**	**Mobile (QCIF)**	**News (QCIF)**	**Mother (QCIF)**
Skip	5651	3072	7044	22341	6929	5284
16×16	8978	9343	16611	26464	13091	10278
16×8	10824	10888	15852	26973	14213	11276
8×16	10291	11230	16782	26929	15161	11828
P8×8	16248	14798	22924	30560	21593	14425
8×8	1885	963	1976	5375	1856	1584
8×4	3549	3024	3972	6546	4389	2969
4×8	3419	3301	4263	6480	4571	3537
4×4	5565	5673	6580	7601	6984	5101

Table 4.2: Detailed R-D cost analysis for different video sequences. Mean values for QP=32 and IP pattern.

Table 4.1 shows the results for the *Paris* video sequence at different QP values and for an IP GOP structure. Part (A) shows the mean values of the R-D cost conditional PDFs for each mode (P8×8 refers to the accumulated cost at sub-MB level); part (B) shows the standard deviations; and part (C) shows the *a priori* probability for each mode. In addition, the R-D cost statistics for each particular sub-MB mode are given in the lower half of the table. Note that the costs at the sub-MB level are proportionally lower than those at the MB level.

Considering these results, some conclusions (that are consistent for other video sequences and formats too) can be drawn:

1. Skip mode exhibits the smallest mean costs. This mode is usually selected in areas of the image that show either constant or no motion and produces low

costs because it saves the transmission of motion-related information.

2. The larger the mode, the smaller the mean and standard deviation of its R-D cost. This behavior is observed at both the MB and sub-MB levels. This can be easily understood since large modes are usually more suitable for either low-detail or stationary regions, which can be represented with very few bits.

3. Rectangular modes (16×8 and 8×16) exhibit similar statistics to each other. The same conclusion is reached at the sub-MB level (8×4 and 4×8 modes).

4. Cost means and deviations generally increase with QP (low qualities). This is due to the fact that, when increasing the QP, the rate term in (3.6) dominates the encoder decisions (to generate lower bit rate); thus, both λ_{mode} and the distortion term of (3.6) increase accordingly, leading to higher global costs.

5. The *a priori* probability of the Skip and 16×16 modes clearly dominates over the remaining ones.

6. The *a priori* probability of the Skip mode increases with QP. Since the weight of the rate term in (3.6) increases with QP, the Skip mode becomes the most likely one.

Table 4.2 shows the mean values of the R-D costs for several well-known video sequences at QP=32. As expected, R-D cost values depend heavily on the specific video sequence.

For illustrative purposes, Figure 4.2 shows the results obtained for the P frames of the video sequence *Foreman* in CIF format with a QP value of 32. The upper part of the figure shows the conditional PDFs for every mode. $J_{8\times8}$ is the accumulated cost of the best 8×8 mode for each sub-MB partition. As can be observed, the individual PDFs (considering those of 16×8 and 8×16 as a whole) are different enough to distinguish the optimal mode. The joint PDF counting all the modes, also depicted in the upper part of Figure 4.2, has been computed as follows:

$$\Pr_J(J_{mode,min}) = \sum_k \Pr_J(J_{mode,k}|k^* = k)\Pr(k^* = k), \tag{4.4}$$

where $J_{mode,min}$ is the minimum R-D cost, corresponding to the optimal coding mode for each MB.

The lower left part of the figure shows the *a priori* probabilities of each mode k being optimal. The lower right part of the figure shows the means and standard deviations of the upper conditional PDFs.

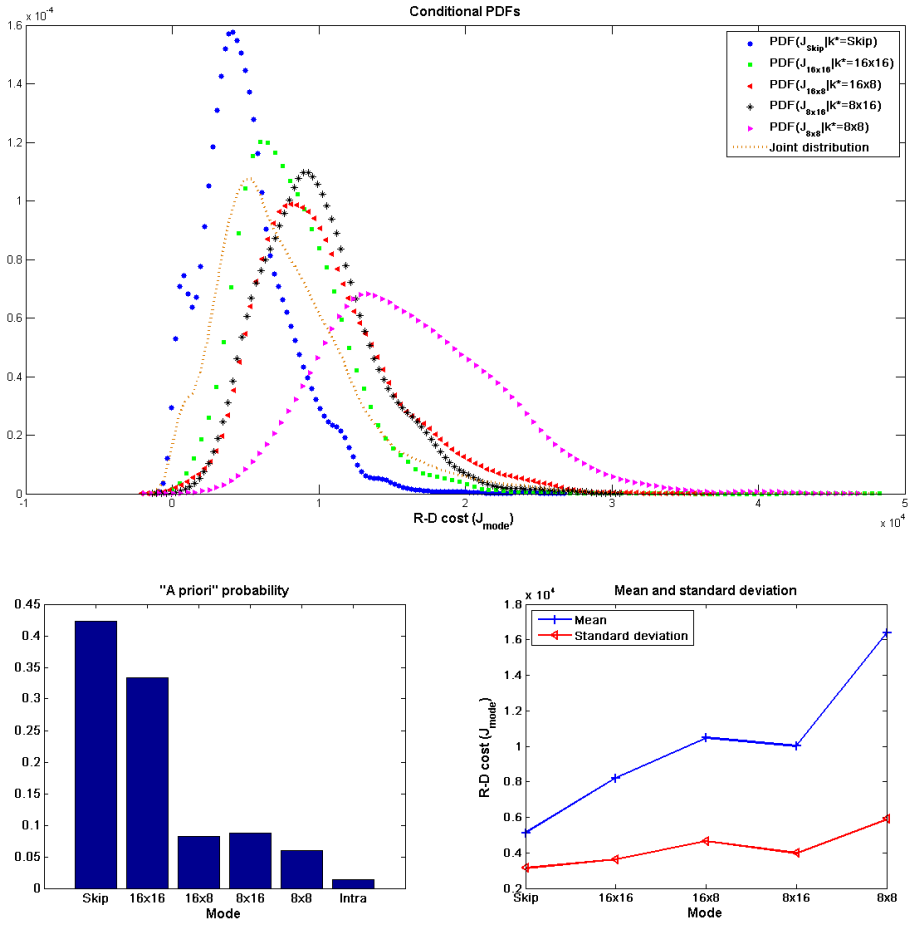


Figure 4.2: Example of the statistical analysis for P frames of the *Foreman* video sequence in CIF format at QP=32. The upper part shows the conditional PDFs for individual modes, $\Pr_J(J_{mode,k}|k^* = k)$, and the joint distribution including all the modes. The lower left part shows the *a priori* probabilities of the individual modes. The lower right part shows the means and standard deviations of the upper conditional PDFs.

4.3.3 Feature Selection

The hypothesis test (4.3) is based on the PDFs, computed according to an observation x, conditioned to each considered hypothesis ($\Pr_x(x|H_i)$), with $i = \{0, 1\}$. Consequently, the selection of this input feature x becomes crucial to the success of the proposed method.

Considering the previous conclusions regarding the R-D costs, it is clear that the R-D cost for a specific mode provides valuable information about the likelihood that this mode is the optimal one. Now, a comprehensive feature selection process is conducted to choose the most appropriate J_{mode} for describing our decision domain, i.e., the R-D cost J_{mode} that produces the most separable PDFs, $\Pr_x(x|H_0)$ and $\Pr_x(x|H_1)$, considering the binary decision process where the hypothesis H_0 entails a low-complexity encoding model (Skip, Inter 16×16, or Intra 16×16) and H_1 entails a high-complexity encoding model (any available mode).

In particular, we seek the most appropriate J_{mode} cost to make an early detection of the MBs that should be encoded as Skip, Inter 16×16, or Intra 16×16 (the modes used in H_0), without causing significant efficiency coding losses. For this purpose, we compute the probability of the $J_{mode,k}$ cost associated with the k partition mode (hereafter simply J_k), when hypothesis H_i, with $i = \{0, 1\}$, is true: $\Pr_J(J_k|H_i)$. In our case, since the modes Skip, Inter 16×16, and Intra 16×16 are assessed for all the MBs and their corresponding J_{mode} costs are always available, we consider the next set of possible costs J_k as candidates for input feature x to our hypothesis testing: J_{Skip}, $J_{Inter16\times16}$, $J_{Intra16\times16}$, and $J_{min(Skip,Inter16)}$, where $min(Skip, Inter16)$ is the minimum cost between the Skip and the Inter 16×16 modes.

To select the most appropriate input feature x out of the considered set of costs, we rely on two different tests: the Bhattacharyya distance and the mutual information (MI). The Bhattacharyya distance measures the distance between two PDFs and, for the Gaussian case, is defined as follows:

$$D_{bhat} = \frac{1}{8}(\mu_2 - \mu_1)^T \left[\frac{\sigma_1^2 + \sigma_2^2}{2}\right]^{-1} (\mu_2 - \mu_1) + \frac{1}{2} ln \frac{\frac{\sigma_1^2+\sigma_2^2}{2}}{\sqrt{|\sigma_1^2 \sigma_2^2|}}, \tag{4.5}$$

where μ_1 and μ_2 are the means and σ_1^2 and σ_2^2 are the variances of the two involved PDFs. In our case, we have to compute the distance between $\Pr_J(J_k|H_0)$ and $\Pr_J(J_k|H_1)$ for every J_k considered and choose as optimal the J_k that maximizes the distance. In other words, the larger the difference between the distributions, the better J_k is as an input feature for the hypothesis testing, since with larger distances it would be easier to distinguish between samples coming from each distribution.

On the other hand, the MI between a set of n random variables $X_1, X_2, ..., X_n$ and the correct decision Y is formulated as:

$$\begin{aligned} MI\left(X_1, X_2, ..., X_n, Y\right) &= h\left(X_1, X_2, ..., X_n\right) - h\left(X_1, X_2, ..., X_n|Y\right) = \\ &= \int\int p_{xy}\left(x_1, x_2, ..., x_n, y\right) \log \frac{p_{xy}(x_1,x_2,...,x_n,y)}{p_x(x_1,x_2,...,x_n)p_y(y)} d\mathbf{x}d\mathbf{y}, \end{aligned} \tag{4.6}$$

where $p_x\left(x_1, x_2, ..., x_n\right)$ is the joint PDF of the n features, $p_y\left(y\right)$ is the marginal PDF of the correct decision Y, $p_{xy}\left(x_1, x_2, ..., x_n, y\right)$ is the joint PDF of the features and the decision, and $h(\cdot)$ is the Shannon differential entropy. Intuitively, MI measures how much the knowledge of a set of features reduces the uncertainty about the correct decision. In our case Y denotes our decision problem, i.e., if an MB is encoded at either low or high complexity, and X denotes the J_k cost (we only have one feature, so $n = 1$). Therefore, $MI(J_k, Y)$ represents the mutual information between the optimal decision and the J_k cost. The higher the MI, the lower the uncertainty about the decision, and the better J_k is as an input feature for the hypothesis testing. In our experiments, we used the estimator described in [Kraskov et al., 2004] to compute the MI.

To select the most suitable feature, we relied on a set of 10 video sequences of different resolutions (4 QCIF, 4 CIF, and 2 HD), and we considered a variety

of quality levels (QP = 24, 28, 32, 36, and 40) with an IP GOP structure. We computed both the Bhattacharyya distance and the MI in all the cases. According to the Bhattacharyya distance, the results achieved are remarkably consistent and in favor of $min(Skip, Inter16)$. When the MI is considered, the results are not so consistent, but again $min(Skip, Inter16)$ turns out to be the most voted. Tables 4.3, 4.4, and 4.5 illustrate these results for three selected examples: *Rush Hour* (HD) at QP 24, *Foreman* (CIF) at QP 32, and *Carphone* (QCIF) at QP 36. In view of the results in these tables, the J_k associated with $min(Skip, Inter16)$, hereafter $J_{Skip,16}$, is the most suitable for our proposal and it is used as input feature in our hypothesis testing (4.3).

	$J_{min(Skip,Inter16)}$	J_{Skip}	$J_{Inter16\times16}$	$J_{Intra16\times16}$
D_{bhat}	0.44	0.04	0.03	0.01
MI	0.20	0.19	0.17	0.10

Table 4.3: D_{bhat} and MI computed for each J_k considered for *Rush Hour* (HD) at QP 24.

	$J_{min(Skip,Inter16)}$	J_{Skip}	$J_{Inter16\times16}$	$J_{Intra16\times16}$
D_{bhat}	0.21	0.10	0.02	0.01
MI	0.14	0.11	0.11	0.09

Table 4.4: D_{bhat} and MI computed for each J_k considered for *Foreman* (CIF) at QP 32.

	$J_{min(Skip,Inter16)}$	J_{Skip}	$J_{Inter16\times16}$	$J_{Intra16\times16}$
D_{bhat}	0.49	0.09	0.02	0.002
MI	0.18	0.10	0.15	0.08

Table 4.5: D_{bhat} and MI computed for each J_k considered for *Carphone* (QCIF) at QP 36.

Figure 4.3 depicts the resulting PDFs for the same examples. The left part of the figure shows $\Pr_J(J_{Skip,16}|H_0)$, in dotted line, and $\Pr_J(J_{Skip,16}|H_1)$, in line with crosses, for the sequence *Rush Hour* (HD) at QP 24; the central part shows the same

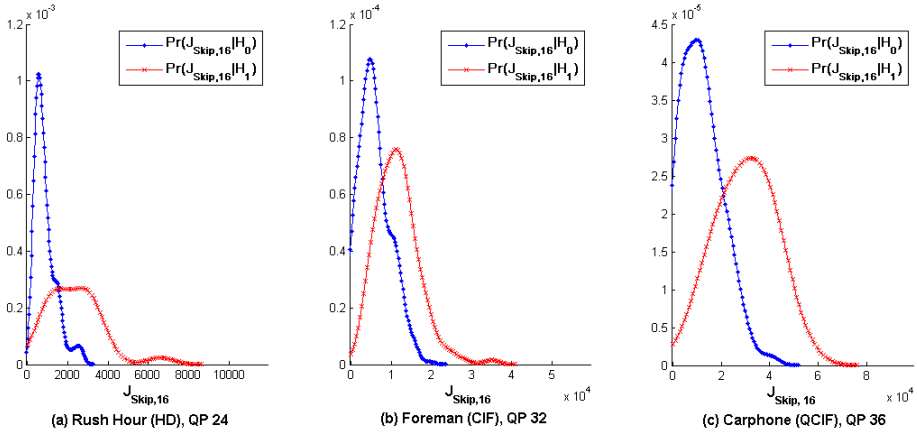


Figure 4.3: Examples of $\Pr_J(J_{Skip,16}|H_0)$ and $\Pr_J(J_{Skip,16}|H_1)$. a) *Rush Hour* (HD) at QP 24; b) *Foreman* (CIF) at QP 32; and c) *Carphone* (QCIF) at QP 36.

PDFs for *Foreman* (CIF) at QP 32; and the right part shows them for *Carphone* (QCIF) at QP 36. As can be observed, the separability of the distributions is enough to make reliable decisions.

Furthermore, as it can be inferred from the previous analysis, the $J_{Skip,16}$ cost is a content-dependent feature that depends on the particular video content and on the QP. Consequently, and following the conclusions of Section 4.3.2, the considered PDFs must be estimated *on the fly* to accurately follow the changing properties of the actual distributions. As it will be proved later, this content-adaptive property is one of the main advantages of our proposal. On the other hand, this approach entails two problems, namely, the computational cost associated with the PDF estimation, and the time to reach the convergence. In order to arrive at a practical solution, the PDFs have been assumed to be Gaussians, so that only their means and standard deviations have to be computed. This hypothesis looks reasonable from the results observed in Figures 4.2 and 4.3 (the same statistical behavior was found for all the sequences and QP values analyzed). The convergence of the mean and standard deviation estimation procedures is studied in following sections.

4.3.4 Content-Adaptive Hypothesis Testing

Once the hypotheses H_0 and H_1 have been defined, the input feature $x = J_{Skip,16}$ selected, and the resulting conditional PDFs $\Pr_J(J_{Skip,16}|H_0)$ and $\Pr_J(J_{Skip,16}|H_1)$ modeled as Gaussian distributions, the LRT defined in (4.3) can be rewritten accordingly:

$$\frac{\exp(\frac{-(J_{Skip,16}-\hat{\mu}_1)^2}{2\hat{\sigma}_1^2})}{\exp(\frac{-(J_{Skip,16}-\hat{\mu}_0)^2}{2\hat{\sigma}_0^2})}\frac{\hat{\sigma}_0^2}{\hat{\sigma}_1^2} \underset{D_0}{\overset{D_1}{\gtrless}} \frac{\hat{P}(H_0)}{\hat{P}(H_1)}\frac{C_{10}}{C_{01}}, \tag{4.7}$$

where $\hat{\mu}_0$ and $\hat{\mu}_1$ are the estimated means of the class conditional PDFs ($\Pr_J(J_{Skip,16}|H_0)$ and $\Pr_J(J_{Skip,16}|H_1)$), respectively; $\hat{\sigma}_0$ and $\hat{\sigma}_1$ are the estimated standard deviations of the same distributions; $\hat{P}(H_0)$ and $\hat{P}(H_1)$ are the estimated *a priori* probabilities of the hypothesis; and the cost associated with correct decisions (C_{00} and C_{11}) are considered to be zero. The parameters of the PDFs, $\hat{\mu}_0$, $\hat{\mu}_1$, $\hat{\sigma}_0$, and $\hat{\sigma}_1$, as well as the *a priori* probabilities $\hat{P}(H_0)$ and $\hat{P}(H_1)$, are estimated *on the fly*, so that the decision process is adapted to the specific video content.

In this Thesis, two different procedures have been studied to carry out the *on the fly* estimation; specifically, an arithmetic and an exponential moving averages. Figure 4.4 shows the estimated mean value of the PDF $\Pr_J(J_{Skip,16}|H_0)$, $\hat{\mu}_0$, when using both kinds of averages. Additionally, the expected mean values per blocks of 50 samples are shown, helping to evaluate the tracking ability of both estimation methods. As it can be observed, the exponential moving average performs better tracking than the arithmetic, maintaining the mean value closer to the expected value, while the arithmetic moving average gets stuck as the number of samples grows. This is due to that the distant samples are less significant than current samples in the exponential procedure.

According to this conclusion, we will use an exponential moving average for the estimation of the PDFs parameters; in particular:

$$\hat{\mu}_i(t) = \alpha\hat{\mu}_i(t-1) + (1-\alpha)J_{Skip,16}(t), \text{ with } i = \{0,1\} \tag{4.8}$$

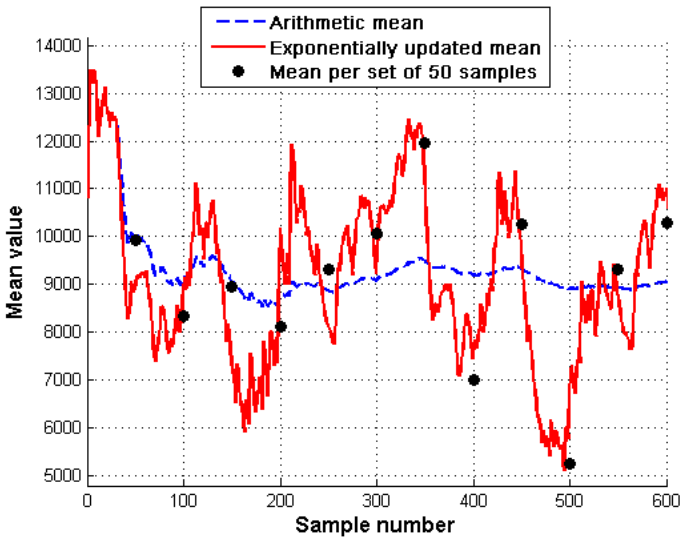


Figure 4.4: Estimation of $\hat{\mu}_0$ with arithmetic and exponential moving averages. *Paris* (CIF) IP, QP 32.

$$\hat{\sigma}_i^2(t) = \beta\hat{\sigma}_i^2(t-1) + (1-\beta)(J_{Skip,16}(t) - \hat{\mu}_i(t))^2, \text{ with } i = \{0,1\}, \tag{4.9}$$

where t denotes a index associated with the time instants for which the H_i hypothesis is selected; $\hat{\mu}_i(t-1)$ and $\hat{\sigma}_i^2(t-1)$ are the estimated mean and variance, respectively, at the instant $(t-1)$; $\hat{\mu}_i(t)$ and $\hat{\sigma}_i^2(t)$ are the estimated mean and variance, respectively, at the instant t; $J_{Skip,16}(t)$ is the cost of the involved MB at the instant t; and α and β are the parameters defining the forgetting factors of the exponentially averaged estimation process. Both α and β were experimentally set to 0.95.

The *a priori* probabilities $\hat{P}(H_0)$ and $\hat{P}(H_1)$ are also estimated *on the fly* by simply counting the number of occurrences of every hypothesis. Additionally, their maximum values are limited to avoid a very unbalanced ratio causing that one hypothesis is always selected.

Finally, it is worth mentioning that the hypothesis test does not begin its operation until a reasonable estimation of all of these parameters is reached; in particular, we establish that the estimation process of the conditional PDFs must have at least 40 samples to start the early selection procedure.

4.3.5 Content-Adaptive Decision Threshold

The most usual expression for the hypothesis test is obtained by taking logarithms in (4.7):

$$-\frac{(J_{Skip,16}-\hat{\mu}_1)^2}{2\hat{\sigma}_1^2}+\frac{(J_{Skip,16}-\hat{\mu}_0)^2}{2\hat{\sigma}_0^2}+ln\frac{\hat{\sigma}_0^2}{\hat{\sigma}_1^2}\gtrless_{D_0}^{D_1} ln(\frac{\hat{P}(H_0)}{\hat{P}(H_1)})+ln(\frac{C_{10}}{C_{01}}). \quad (4.10)$$

Furthermore, to simplify (4.10), hereafter we will refer to the left and right sides of this equation as follows:

$$\theta \gtrless_{D_0}^{D_1} \eta+\epsilon, \quad (4.11)$$

where θ is the left part of the inequality in (4.10), η refers to the logarithm of the *a priori* probability ratio $\Big(ln(\hat{P}(H_0)/\hat{P}(H_1))\Big)$, and ϵ refers to the logarithm of the cost ratio ($ln(C_{10}/C_{01})$).

To control the complexity, we propose to act on ϵ (cost ratio) in (4.11). By acting on ϵ, we are varying the threshold according to which the hypothesis testing decides whether an MB is encoded using the low-complexity model (D_0) or the high-complexity model (D_1). As can be seen in the inequality in (4.11), the larger the ϵ, the higher the number of low-complexity encoded MBs.

It should be noticed that by acting on ϵ we are actually modifying the relative importance of C_{01} and C_{10}. When low complexity is required, the cost of deciding the high complexity hypothesis when the other was the correct one is large. In such a case, C_{10} takes a high value and, consequently, ϵ also takes a high value. In contrast, when a high value of complexity is acceptable, the complexity control algorithm should focus on coding efficiency. In this case, deciding low complexity when high complexity was the correct decision becomes more relevant; C_{01} takes a high value, and ϵ a low value. In summary, high values of C_{10} promote complexity saving, while high values of C_{01} benefit coding efficiency.

The goal of the complexity control is to act on ϵ to achieve a certain Target Complexity (TC). This TC is expressed as a percentage of the full complexity, i.e.,

$TC = 100$ means that the target complexity is that of the full mode evaluation, or $TC = 20$ means that the target complexity is 20% of the full mode evaluation. This TC value could be given according to one or several parameters, as the current battery level in a mobile device, the buffer occupancy in rate-controlled transmission application, or the available CPU resources in non-dedicated multi-task systems.

TC is converted into an equivalent parameter that is directly managed by the proposed algorithm: the number of MBs encoded in low complexity mode, MB_{low}. Actually, each time the hypothesis testing decides D_0, a low complexity MB is encoded. In this way, if TC is low, MB_{low} should be high and vice-versa.

Given a target complexity TC, MB_{low} is computed as follows. Let us define μ_{high} and μ_{low} as the average time spent for encoding an MB at high- or low-complexity, respectively. These two parameters are computed by simply averaging the real encoding time spent on each type of MB over several MBs, and are initialized using the first high- and low-complexity samples, respectively. Let us define now the target time that should be spent per frame, TT, to meet TC:

$$TT = time_{per-frame-full} \frac{TC}{100}, \tag{4.12}$$

where $time_{per-frame-full}$ denotes the time spent encoding a whole frame at full complexity. We rewrite the previous equation by expressing the time per frame as a function of the number of MBs in a frame, $MB_{per-frame}$:

$$TT = (\mu_{high} MB_{per-frame}) \frac{TC}{100}. \tag{4.13}$$

Likewise, the target time TT can be expressed in terms of the number of MBs encoded at high complexity, MB_{high}, the number of MBs encoded at low complexity, MB_{low}, and the corresponding average coding times per MB, μ_{high} and μ_{low}:

$$TT = (\mu_{high} MB_{high}) + (\mu_{low} MB_{low}). \tag{4.14}$$

When equations (4.13) and (4.14) are combined, the number of MB encoded at low complexity MB_{low} can be easily found as a function of TC:

$$MB_{low} = \frac{(\mu_{high} MB_{per-frame})\left(1 - \frac{TC}{100}\right)}{\mu_{high} - \mu_{low}}. \tag{4.15}$$

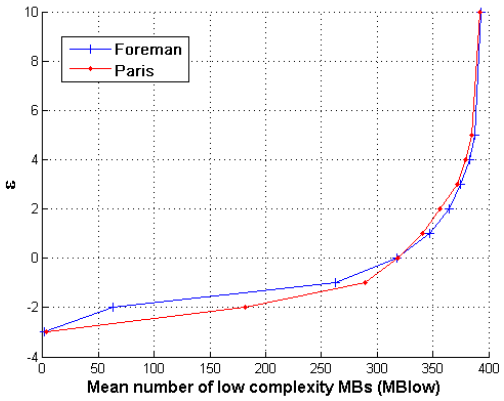


Figure 4.5: An illustration of the relationship between the number of MBs encoded at low complexity MB_{low} and the threshold ϵ for *Paris* and *Foreman*.

Once TC is converted into MB_{low}, we can tackle the problem of selecting a specific value for the threshold ϵ so that a given MB_{low} is met. The relationship between ϵ and MB_{low} has been studied experimentally. Figure 4.5 illustrates the result by means of two examples. One of the curves is derived from *Paris* and the other from *Foreman*, both with CIF resolution, at QP=28. It can be observed that MB_{low} (the number of early stops) increases with ϵ until saturation. The saturation of the curve indicates that $MB_{low} = MB_{per-frame}$, i.e., all the MBs (396 for the CIF sequences of our example) are encoded at low complexity, reaching the lowest complexity level achievable by the proposed method.

It is worth noting that the number of early stops obtained for a given ϵ actually depends on the video content. For example, $\epsilon = -2$ produces $MB_{low} = 182$ for *Paris* and $MB_{low} = 63$ for *Foreman*. Furthermore, the differences between curves are more significant for low values of ϵ due to the low slope of the curve. Moreover, the statistics in (4.10) are time-variant; therefore, fixing a specific value of ϵ would produce meaningful differences in the number of early stops MB_{low} from frame to frame. So, the number MB_{low} produced by a value of ϵ is dependent of the video content, which varies among different sequences and inside the same sequence.

Because of these reasons, ϵ must be adjusted *on the fly* to follow the time-variant statistics and meet the target MB_{low}. Specifically, we propose to update ϵ in a frame-by-frame basis by means of a feedback algorithm, as follows:

$$\epsilon_f = \epsilon_{f-1} + (\nu \times \Delta MB_{low}), \tag{4.16}$$

where ϵ_f and ϵ_{f-1} are the thresholds applied to the $f-th$ and $(f-1)-th$ frames, respectively; ΔMB_{low} is the difference between the MB_{low} target for the $f-th$ frame and the actual MB_{low} obtained for the $(f-1)-th$ frame; and ν is a parameter experimentally determined as a function of ΔMB_{low} and the frame size.

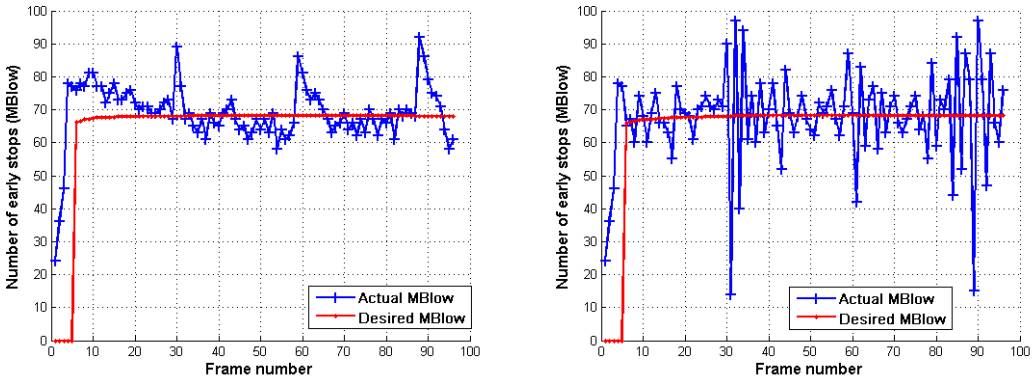
The ν value allows for choosing an application-specific operating point that properly balances the adaptation speed versus the amplitude of the oscillations around the target complexity. If a high value of ν is used, the target time per frame, TT, will be reached faster, but a larger oscillation around this TT will be observed, and vice-versa. Figure 4.6 illustrates this behavior for *Mobile* (QCIF) at QP 28. The resulting time evolution of MB_{low} (the number of MBs encoded at low complexity) is shown for two values of ν. As can be seen, for $\nu = 0.005$ (left part of the figure), some frames are needed to reach the desired value of MB_{low}, but the oscillations around it are moderated. In contrast, for $\nu = 0.1$ (right part of the figure), the desired value of MB_{low} is reached much faster, but at the expense of larger oscillations.

To properly manage this trade-off, the value of ν is varied adaptively according to the magnitude of ΔMB_{low}: the higher ΔMB_{low}, the higher ν. In this manner, when encoding time is far from TT, ϵ is adapted faster, and vice-versa. Furthermore, different ν values are used for each spatial resolution (QCIF, CIF, and HD), specifically:

QCIF: $|\Delta MB_{low}| > 20 \Rightarrow \nu = 0.05$; $|\Delta MB_{low}| < 5 \Rightarrow \nu = 0$; other case: $\nu = 0.05$.

CIF: $|\Delta MB_{low}| > 50 \Rightarrow \nu = 0.025$; $|\Delta MB_{low}| < 5 \Rightarrow \nu = 0$; other case: $\nu = 0.01$.

HD: $|\Delta MB_{low}| > 80 \Rightarrow \nu = 0.001$; $|\Delta MB_{low}| < 5 \Rightarrow \nu = 0$; other case: $\nu = 0.0005$.



(a) $\nu = 0.005$. (b) $\nu = 0.1$.

Figure 4.6: Illustration of the role of the ν parameter, which controls the balance between complexity adaptation speed and oscillation amplitude. This results have been obtained for *Mobile* (QCIF) at QP 28.

4.3.6 Summary of the Algorithm

Algorithm 1 summarizes the complete algorithm.

Algorithm 1 Proposed complexity control algorithm.

Require: F: number of frames.
Require: M: number of MBs in a frame.
  for $\forall f_i \in F$ **do**
    Calculate MB_{low} based on the mean time measures and the demanded encoding time (4.15).
    Calculate the threshold ϵ based on the feedback algorithm (4.16).
    for $\forall m_i \in M$ **do**
      Evaluate Skip, Inter 16x16, and Intra 16x16 modes.
      Calculate the input feature to the hypothesis testing J_{Skip16}.
      Apply the hypothesis testing (4.11).
      if $\theta < \eta + \epsilon$ **then**
        Decide the best mode between Skip, Inter 16x16, and Intra 16x16.
      else
        Calculate all remaining modes.
        Decide the best mode.
      end if
      Update μ_{high} and μ_{low}, and statistics in (4.10).
    end for
  end for

4.4 Experimental results

4.4.1 Experimental Protocol

To assess the performance of the proposed method, it was integrated into the H.264/AVC reference software JM10.2. The main test conditions were selected according to the recommendations of the JVT [G.Sullivan, 2001], namely: main profile, ±32 pixel search range for QCIF and CIF and ±64 pixels for HD, 5 reference frames, Hadamard transform, CABAC, and RDO. The experiments were conducted using an IPPP GOP pattern, five QP values (24, 28, 32, 36 and 40), and 100 frames per sequence. Table 4.6 summarizes these conditions.

The experiments involved a large set of sequences of different resolutions covering a wide variety of contents. These sequences are listed in Tables 4.7, 4.8, and 4.9 for QCIF, CIF, and HD resolutions, respectively.

To evaluate the capability of the algorithm to meet a certain target complexity

Coding options	
Profile	Main
RD Optimization	Enabled
Use Hadamard	Enabled
Symbol Mode	CABAC
Search Range (QCIF, CIF)	± 32
Search Range (HD)	± 64
QP	$24, 28, 32, 36, 40$
Number of Reference Frames	5
Frames to be encoded	100
GOP pattern	IPPP

Table 4.6: Test conditions.

TC, a measurement of computational time saving TS was calculated as follows:

$$TS = \frac{Time(JM10.2) - Time(Proposed)}{Time(JM10.2)} \times 100. \tag{4.17}$$

Thus, the higher the measured TS, the lower the reached complexity. In particular, the proposed algorithm was assessed for seven different target complexities, $TC(\%) = \{80, 70, 60, 50, 40, 30, 20\}$.

Furthermore, to evaluate the coding efficiency losses incurred by the proposed method due to the complexity control, average bit rate increments ($BDBR$), with respect to reference software, were calculated as described in [G.Bjontegaard, 2001].

4.4.2 Performance Assessment

Tables 4.7, 4.8, and 4.9 show the results for QCIF, CIF, and HD resolutions, respectively. Specifically, for each of the TCs considered, the mean values of $TS(\%)$ and $BDBR(\%)$ across the five considered QP values are given. Furthermore, the last row of each table shows the average results for all the sequences.

As can be observed, the achieved complexity was very close to TC. Therefore, the method is successful in fulfilling the main goal of having a precise complexity control. Moreover, the coding efficiency was maintained very close to that of the reference implementation when medium or high TCs were sought. Obviously, when low TCs were demanded, these were achieved in exchange for more significant losses

in coding efficiency.

It is worth mentioning that, exceptionally, bit rate reductions were found. These unexpected results were achieved because the encoder decisions are sub-optimal in the sense that they are made assuming independence between MBs. Thus, in some cases, a decision that is not locally- optimal (in the sense that only explores a subset of modes) could produce better overall performance.

TC	20%		30%		40%		50%		60%		70%		80%	
Sequence	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)
Akiyo	76.1	5.1	68.2	1.0	57.7	0.4	48.4	0.3	39.5	0.2	30.2	0.0	22.0	0.0
Bridge close	75.3	3.3	65.5	2.0	55.8	1.0	46.4	0.5	37.9	0.3	28.9	0.3	20.1	0.2
Bridge far	72.0	1.1	64.6	0.8	54.8	0.5	44.1	0.3	37.7	0.2	28.8	0.1	21.1	0.1
Carphone	79.8	11.3	67.9	5.6	57.1	2.8	46.7	1.7	37.2	0.9	28.0	0.0	19.0	0.3
Claire	77.9	5.4	65.0	0.6	54.6	0.2	45.0	−0.2	36.4	−0.1	28.1	−0.1	20.6	−0.3
Coastguard	82.1	9.8	74.8	5.4	62.2	3.2	50.4	1.9	39.9	1.3	30.5	0.8	21.5	0.4
Container	76.0	6.9	66.7	2.7	55.4	1.2	44.4	0.3	35.2	0.1	26.3	0.2	18.2	0.1
Foreman	82.2	17.7	67.7	8.8	56.5	4.7	45.8	2.5	36.4	1.4	27.7	0.6	20.2	0.1
Grandma	77.6	5.7	69.8	1.7	58.4	0.8	48.3	0.5	36.7	0.3	27.6	0.2	18.4	−0.2
Hall	73.5	5.6	65.2	1.13	56.5	0.9	47.0	0.1	38.3	0.2	30.5	−0.1	22.4	0.1
Highway	75.8	12.0	64.3	4.6	53.1	2.5	42.6	1.3	33.5	1.1	26.4	1.0	19.2	0.9
Miss America	74.5	4.2	64.8	1.3	53.7	0.2	42.0	0.0	33.5	−0.1	25.6	−0.3	18.8	−0.4
Mobile	83.8	15.5	71.3	10.2	60.0	7.3	49.5	5.2	39.2	3.5	29.9	2.4	20.7	1.4
M&D	78.2	6.9	67.1	2.2	54.6	1.0	42.7	0.2	32.3	0.3	24.3	−0.2	16.7	0.0
News	76.5	8.3	67.1	2.8	55.8	0.9	45.9	0.3	37.2	0.3	29.1	0.2	20.8	0.3
Salesman	79.1	9.0	71.0	2.8	59.9	0.9	49.4	0.0	39.1	0.1	29.5	−0.1	20.1	0.0
Silent	77.5	8.6	68.4	2.8	58.8	1.4	49.4	1.0	40.0	0.7	31.9	0.5	23.5	0.0
Suzie	78.8	10.7	69.8	5.5	55.6	3.0	44.5	1.7	33.6	0.8	24.1	0.3	16.4	0.3
Average	**77.6**	**8.2**	**67.7**	**3.5**	**56.7**	**1.8**	**46.3**	**1.0**	**36.9**	**0.6**	**28.2**	**0.3**	**20.0**	**0.2**

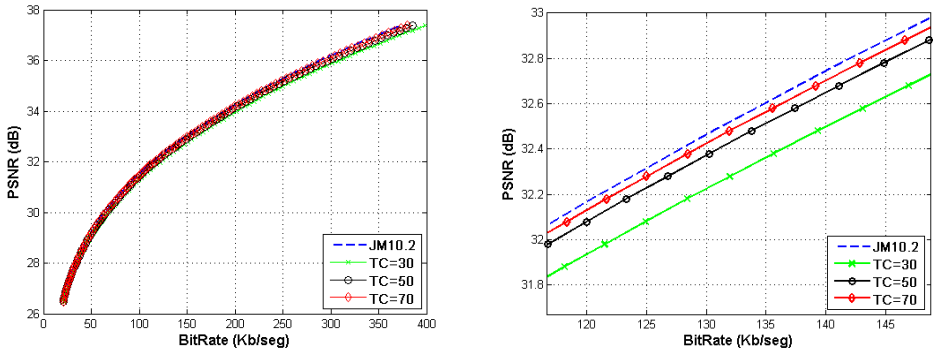
Table 4.7: Performance evaluation of the proposed complexity control algorithm in H.264 relative to JM10.2 for QCIF sequences. TS stands for time saving and $BDBR$ stands for bit rate increment.

TC	20%		30%		40%		50%		60%		70%		80%	
Sequence	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)
Akiyo	76.8	3.5	66.6	0.7	55.6	0.0	46.0	0.0	37.8	0.0	30.5	0.0	22.2	0.0
Bus	82.1	17.9	70.5	8.0	60.3	4.4	51.0	2.9	42.6	2.5	34.0	2.4	24.5	1.4
Coastguard	83.5	6.2	74.2	3.9	62.2	2.6	50.3	2.0	40.8	1.4	32.6	1.1	22.4	0.5
Container	79.0	5.2	70.1	1.9	59.4	0.5	48.7	0.3	38.8	0.2	30.0	0.0	21.8	−0.1
Football	80.5	21.7	67.6	13.7	53.8	6.7	42.8	2.9	32.7	1.1	24.2	0.7	16.2	0.3
Foreman	80.5	15.2	68.7	5.5	57.9	3.2	47.9	1.7	41.5	2.1	35.1	2.3	23.8	0.8
Garden	82.7	16.8	71.0	11.6	54.7	5.9	42.4	3.6	28.7	2.0	17.6	0.8	9.7	0.3
Highway	75.4	8.5	65.1	3.7	51.5	1.4	42.1	0.6	35.7	0.8	31.2	1.3	20.4	0.3
Mobile	81.0	16.9	67.4	10.6	54.2	6.9	42.6	4.5	32.8	2.9	23.9	1.9	14.3	0.7
M&D	78.9	3.8	66.8	0.9	54.5	0.3	42.6	0.2	33.0	−0.2	24.8	−0.2	17.6	−0.1
News	76.8	6.7	66.6	2.5	56.3	1.0	46.4	0.4	39.1	0.3	32.6	0.3	22.0	0.1
Paris	79.6	14.8	64.8	4.6	54.5	2.0	45.6	0.8	38.5	1.1	31.4	1.1	22.6	0.3
Silent	79.5	6.5	70.0	2.0	60.4	1.3	51.5	0.8	43.1	0.8	35.0	0.7	24.5	0.3
Stefan	76.2	13.9	67.2	10.0	54.8	6.5	40.6	3.1	32.0	1.6	24.5	0.9	16.5	0.6
Tempete	83.3	11.1	69.1	7.0	57.0	4.9	45.9	3.2	37.2	2.2	32.3	1.8	21.3	1.0
Waterfall	82.1	8.0	72.9	3.5	62.0	1.8	52.1	1.4	43.9	0.9	36.3	0.6	25.5	0.5
Average	**79.9**	**11.0**	**68.7**	**5.6**	**56.8**	**3.1**	**46.2**	**1.8**	**37.4**	**1.2**	**29.8**	**1.0**	**20.3**	**0.4**

Table 4.8: Performance evaluation of the proposed complexity control algorithm in H.264 relative to JM10.2 for CIF sequences. TS stands for time saving and $BDBR$ stands for bit rate increment.

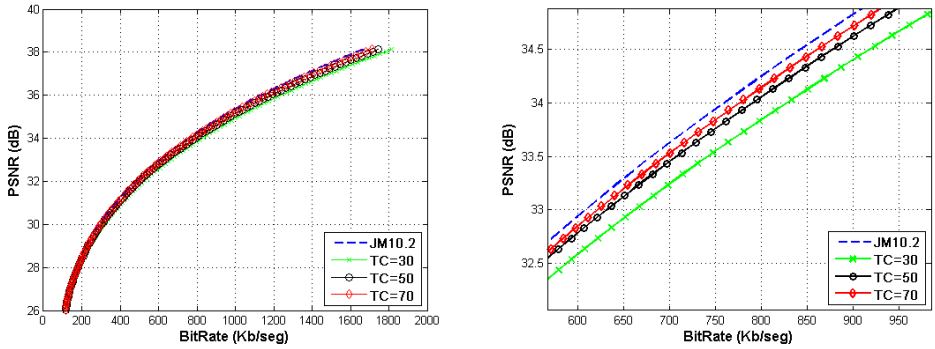
TC	20%		30%		40%		50%		60%		70%		80%	
Sequence	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)
Blue Sky	63.8	2.2	62.9	1.9	55.7	1.4	46.2	0.8	36.7	0.5	28.3	0.4	19.7	0.3
Pedestrian	75.7	5.4	63.8	2.4	52.6	1.1	43.0	0.7	34.6	0.4	26.9	0.3	19.6	0.2
Riverbed	82.0	12.4	72.6	10.0	61.0	7.2	50.3	5.1	40.6	3.6	31.5	2.5	23.2	1.7
Rush Hour	77.7	5.4	66.7	2.6	55.7	1.3	46.2	0.7	37.9	0.4	30.0	0.2	22.4	0.2
Station2	78.4	2.6	71.6	0.9	61.4	0.3	51.6	0.2	42.3	0.1	33.0	0.0	23.4	0.2
Sunflower	76.2	1.8	67.5	1.3	58.2	0.9	49.7	0.5	41.6	0.5	33.4	0.2	25.0	0.2
Tractor	80.6	9.1	70.0	3.8	59.9	1.9	49.9	1.3	40.8	0.8	32.9	0.6	24.3	0.5
Average	**76.4**	**5.6**	**67.9**	**3.3**	**57.8**	**2.0**	**48.1**	**1.3**	**39.2**	**0.9**	**30.9**	**0.6**	**22.5**	**0.5**

Table 4.9: Performance evaluation of the proposed complexity control algorithm in H.264 relative to JM10.2 for HD sequences. TS stands for time saving and $BDBR$ stands for bit rate increment.



(a) R-D performance. (b) Zoom of the Figure on the left.

Figure 4.7: R-D performance for a representative subset of the target complexities considered. *Coastguard* at QCIF resolution.

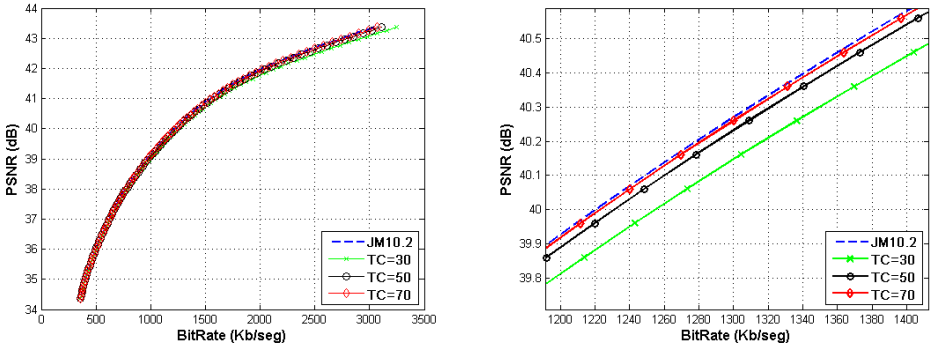


(a) R-D performance. (b) Zoom of the Figure on the left.

Figure 4.8: R-D performance for a representative subset of the target complexities considered. *Tempete* at CIF resolution.

To illustrate how the coding efficiency depends on TC, Figures 4.7, 4.8, and 4.9 show the R-D performance for *Coastguard* (QCIF), *Tempete* (CIF), and *Rush hour* (HD) for $TC(\%) = \{30, 50, 70\}$ (not all the TCs are depicted to make the graph clearer). The left part of each figure presents the complete R-D curves, while the right part presents a zoom of a selected area. As can be observed, the coding efficiency is very close to that of the reference software for high and medium TCs and slightly degrades as the TC decreases.

Although the results in terms of objective R-D measurements are good, we also



(a) R-D performance. (b) Zoom of the Figure on the left.

Figure 4.9: R-D performance for a representative subset of the target complexities considered. *Rush Hour* at HD resolution.

checked that the proposed method does not have negative effects on the subjective quality. To this end, we carefully watched some of the resulting encoded sequences and concluded that there are not perceptual differences with respect to those generated by the reference encoder. Moreover, we labeled the MBs according to the complexity level assigned by the algorithm (low or high) to visually check whether its decisions were as expected. Figure 4.10 shows an illustrative example where the encoder must comply with a tough complexity constraint ($TC = 30$). As can be observed, only a few MBs are encoded with high complexity (light-colored in the figure) and are those related to moving parts.

Moreover, the proposed algorithm was assessed in comparison with the complexity control algorithm proposed in [da Fonseca and de Queiroz, 2009]. Table 4.10 shows the average results achieved by the compared algorithms for several target complexities ($TC(\%) = \{80, 70, 60, 50, 40, 30, 20\}$). In particular, for each one of the image resolutions considered (QCIF, CIF, and HD), an average result was computed taking into account the five QP values and all the test video sequences. As can be seen, for low complexities (20, 30, and 40), the proposed algorithm generates a complexity closer to the target. The same happens for high complexities (70 and 80), where the algorithm in [da Fonseca and de Queiroz, 2009] generates lower complexities than those actually demanded (because it works by selecting a subset of modes

Figure 4.10: Illustration of the decisions made by the proposed algorithm. For a tough target complexity, *Paris* (CIF) with $TC = 30$, we have highlighted those MBs encoded with high complexity. As expected, in general, these MBs belong to moving parts.

and, sometimes, this procedure does not allow for finer complexity control), usually in exchange for a higher increment of bit rate. Furthermore, in general, the proposed algorithm produces significantly lower bit rate increments for the same TC.

To gain an insight into the differences between the performance of the compared algorithms, some graphical examples are shown for several representative sequences. In particular, we show the bit rate increments of the compared algorithms with respect to the reference software as a function of the computational TS. Obviously, for higher TSs, the losses in coding efficiency and, consequently, the bit rate increments are more relevant. Figure 4.11 shows these results for two QCIF sequences, *Coastguard* and *Mother & Daughter*; Figure 4.12 shows the results for two CIF sequences, *Foreman* and *Waterfall*; and Figure 4.13 shows the results for two HD sequences, *Pedestrian* and *Rush Hour*. As can be observed, the proposed algorithm clearly outperformed that proposed in [da Fonseca and de Queiroz, 2009], especially for high computational TSs, where the bit rate increment generated by the proposed algorithm was significantly lower.

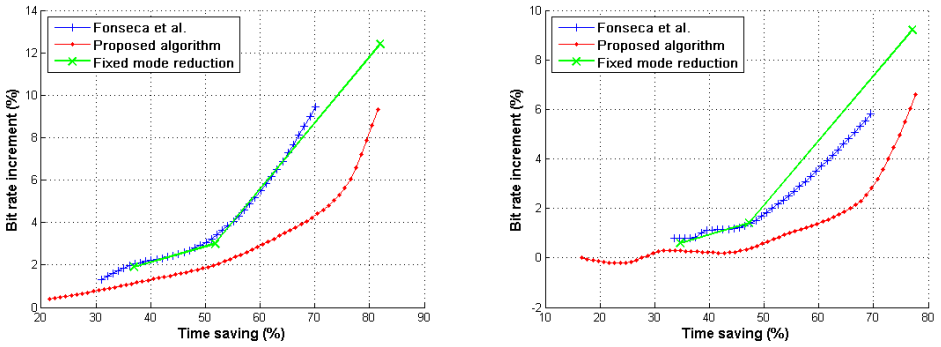
To provide an additional reference, we also compared the proposed algorithm with a fixed mode reduction, i.e., a method that simply explores a predetermined subset of modes. Specifically, we tested three different subsets of Inter modes (Intra modes are always available), namely:

- Skip and Inter 16×16;
- Skip, Inter 16×16, Inter 16×8, and Inter 8×16; and
- Skip, Inter 16×16, Inter 16×8, Inter 8×16, and Inter 8×8.

The results achieved by this method have been added to Figures 4.11, 4.12, and 4.13. In particular, each subset of modes generates a (*Bit rate increment*, *Time saving*) point in these figures (these points have been linked by straight lines to improve visualization). As can be observed, the proposed method achieved better performance for QCIF and CIF resolutions, especially for high TSs. On the other hand, for HD resolution, the results were slightly better for the fixed mode reduction method for low TSs. This last result was expected since the impact on the R-D performance of the small modes (8×4, 4×8, and 4×4) is not significant for HD, and the proposed method explores all of them for high-complexity MBs. However, with higher TSs the results are very similar or better with our proposal as the impact of removing bigger partitions is more relevant. Finally, although this fixed mode reduction is provided as an alternative benchmark, it should be noticed that, actually, it is not a complexity control algorithm (a fixed subset of modes are explored in all the MBs and, therefore, the encoder is not capable of adapting to any target complexity).

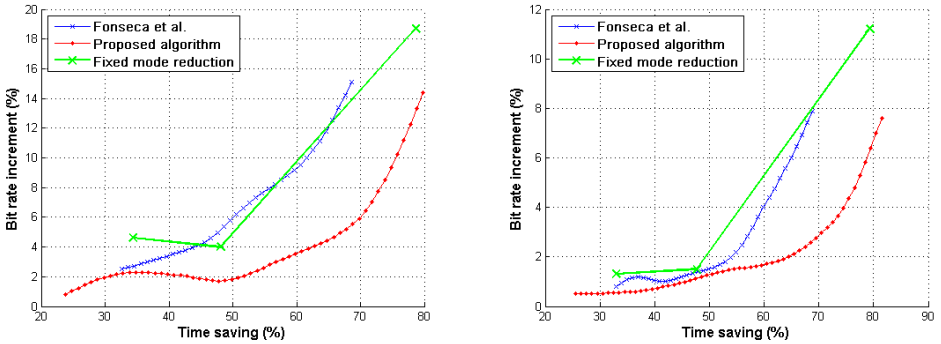
TC	**20%**		**30%**		**40%**		**50%**		**60%**		**70%**		**80%**	
Sequence	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)	**TS** (%)	**BDBR** (%)
Proposed QCIF	77.6	8.2	67.7	3.5	56.7	1.8	46.3	1.0	36.9	0.6	28.2	0.3	20.0	0.2
[da Fonseca and de Queiroz, 2009] QCIF	70.1	7.6	61.5	5.2	52.7	3.6	45.7	2.4	39.8	2.0	35.9	1.4	32.0	1.1
Proposed CIF	79.9	11.0	68.7	5.6	56.8	3.1	46.2	1.8	37.4	1.2	29.8	1.0	20.3	0.4
[da Fonseca and de Queiroz, 2009] CIF	69.9	10.4	60.6	6.8	52.6	4.6	46.1	3.2	39.8	2.4	33.5	1.6	27.4	1.1
Proposed HD	76.4	5.6	67.9	3.3	57.8	2.0	48.1	1.3	39.2	0.9	30.9	0.6	22.5	0.5
[da Fonseca and de Queiroz, 2009] HD	70.2	7.0	62.8	4.2	54.7	2.2	47.6	1.0	41.4	0.5	39.2	0.4	38.4	0.4

Table 4.10: Performance evaluation of the proposed complexity control algorithm in H.264 in comparison with [da Fonseca and de Queiroz, 2009]. Average results. TS stands for time saving and $BDBR$ stands for bit rate increment.



(a) Coastguard (QCIF). (b) Mother & Daughter (QCIF).

Figure 4.11: Performance evaluation of the proposed complexity control method in H.264 in comparison to that in [da Fonseca and de Queiroz, 2009] and to that of a fixed mode reduction for two representative QCIF sequences. The graphs show bit rate increment as a function of the computational time saving.

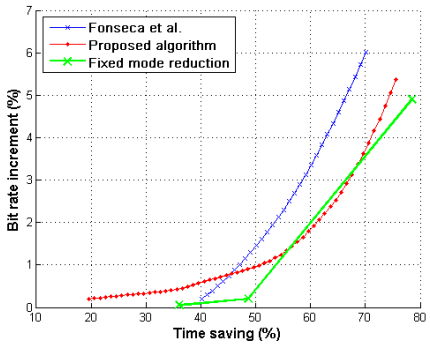


(a) Foreman (CIF). (b) Waterfall (CIF).

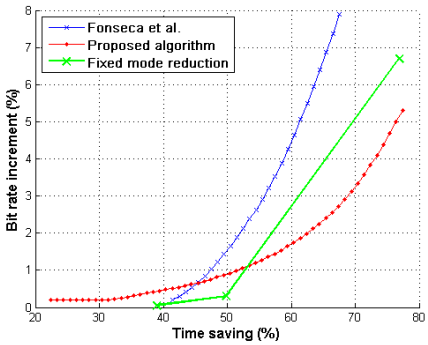
Figure 4.12: Performance evaluation of the proposed complexity control method in H.264 in comparison to that in [da Fonseca and de Queiroz, 2009] and to that of a fixed mode reduction for two representative CIF sequences. The graphs show bit rate increment as a function of the computational time saving.

4.4.3 Illustrations of the algorithm convergence properties

Since the capability to adapt to a time-variant complexity target and to the video content is one of the main goals of the proposed algorithm, some illustrations regarding the algorithm convergence properties are in order.



(a) Pedestrian (HD).



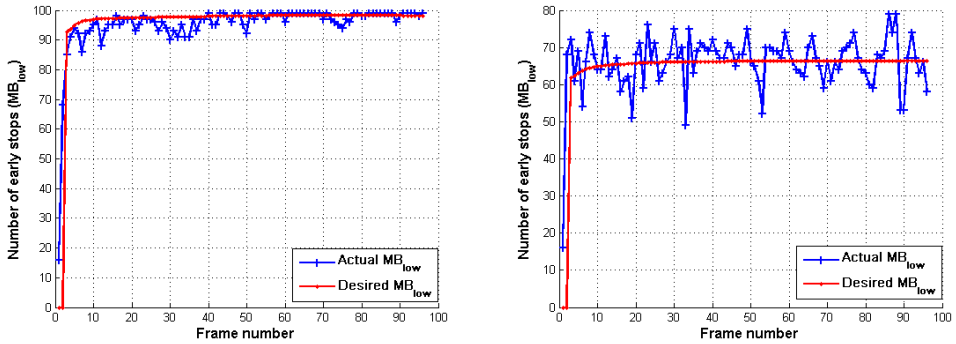
(b) Rush Hour (HD).

Figure 4.13: Performance evaluation of the proposed complexity control method in H.264 in comparison to that in [da Fonseca and de Queiroz, 2009] and to that of a fixed mode reduction for two representative HD sequences. The graphs show bit rate increment as a function of the computational time saving.

First, we provide two graphical examples of the capability of the algorithm to converge to a certain TC. Specifically, Figure 4.14 illustrates, for *Carphone* (QCIF) at QP = 28, how the number of low-complexity MBs evolves with time (frame number) for two different TCs: 20 (Figure 4.14a) and 50 (Figure 4.14b). As can be observed, when TC was set to a low value, 20 in Figure 4.14a, the actual number of early stops (MB_{low}) reached a value very close to the desired one in just a few frames. Furthermore, the variance with respect to the desired value was low. When TC was set to a higher value, 50 in Figure 4.14b, the convergence time was again very small, but in this case the variance around the desired value of MB_{low} was higher. A very similar behavior was observed for almost all the sequences.

Second, Figure 4.15 shows two illustrative examples of a time-variant TC for *Paris* (CIF) at QP=28. On the left part of the figure, we illustrate the behavior of the proposed algorithm when TC changed from 50 to 20 at frame 50. On the right part of the figure, two changes happened: TC went from 20 to 50 at frame 25 and to 30 at frame 50. As shown, the proposed algorithm was able to reach the desired complexity quickly even when fast changes in TC happened.

Finally, to provide a more solid proof of the convergence properties of the algorithm than the previous illustrative examples, we computed average results for



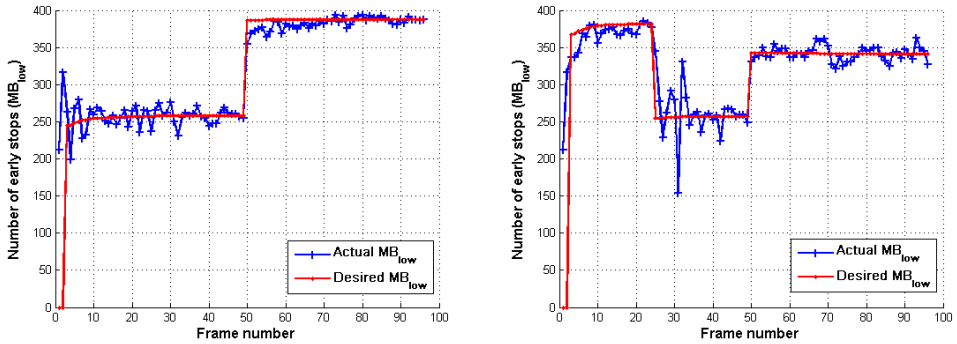
(a) Time evolution of MB_{low} for $TC = 20$. (b) Time evolution of MB_{low} for $TC = 50$.

Figure 4.14: Illustrative examples of the algorithm convergence properties for *Carphone* (QCIF) at QP 28.

	$TC = 20$		$TC = 50$		$TC = 80$	
Sequence	**Desired** MB_{low}	**Actual** MB_{low}	**Desired** MB_{low}	**Actual** MB_{low}	**Desired** MB_{low}	**Actual** MB_{low}
Carphone QP 28 (QCIF)	97	96	66	66	34	34
Container QP 32 (QCIF)	99	98	68	69	34	35
M&D QP 36 (QCIF)	99	94	66	64	32	36
Akiyo QP 28 (CIF)	396	387	271	271	139	139
Mobile QP 36 (CIF)	392	388	261	260	132	131
Silent QP 40 (CIF)	396	394	281	282	141	141
Pedestrian QP 28 (HD)	3528	3453	2368	2377	1189	1191

Table 4.11: Assessment of the convergence properties of the proposed algorithm.

several sequences covering all the image resolutions considered. Specifically, Table 4.11 shows, for some listed sequences and three different target complexities ($TC(\%) = \{20, 50, 80\}$), the actual value of MB_{low} and the desired value of MB_{low} averaged over all the encoded frames. It is worth noticing that these measurements are totally independent of the implementation. These results allow us to conclude that, on average, the proposed algorithm is able to reach TC with a remarkable precision.



(a) Time evolution of MB_{low} for a time-variant TC, which changes from 50 to 20 at frame 50.

(b) Time evolution of MB_{low} for a time-variant TC, which changes from 20 to 50 at frame 25 and to 30 at frame 50.

Figure 4.15: Illustrative examples of the algorithm convergence for a time-variant TC, for *Paris* (CIF) at QP 28.

4.5 Conclusions

In this chapter we have proposed a novel algorithm to control the complexity of an H.264/AVC encoder.

The proposed method relies on the application of a hypothesis testing to meet a target complexity with minimum losses in coding efficiency. Assuming Gaussian distributions, the hypothesis testing paradigm allows us to formulate the problem in a simple form that depends on some statistics that can be estimated *on the fly*. As a result, we have shown that the proposed algorithm fulfills the desired requirements: low miss-adjustment error with respect to the target complexity, capability to adapt to a time-variant complexity target and to the video content, and capability to operate on a large dynamic range of target complexities and image resolutions. Furthermore, the proposed algorithm is computationally simple.

To assess its performance, the proposed algorithm was implemented on the reference software JM10.2. The experimental evaluation was carried out on a large set of sequences of several spatial resolutions, and a comprehensive set of potential target complexities. The results obtained allow us to conclude that the proposed algorithm can reach any target complexity with remarkable precision, adapt to time-variant target complexities, and work properly with any spatial resolution, having negligi-

ble bit rate increments for high and medium complexities and acceptable bit rate increments for very low complexities. When compared with the complexity control method in [da Fonseca and de Queiroz, 2009], the proposed method was able to reach complexities closer to the target and to provide a better trade-off between complexity reduction and coding efficiency, especially for low and medium target complexities.

Chapter 5

Complexity Control in HEVC

In this chapter we present an effective complexity control algorithm for HEVC. Our proposal is based on a hierarchical approach. An early termination condition is defined at every CU depth to determine whether subsequent CU depths should be explored. The actual encoding times are also considered to satisfy the target complexity in real time. Moreover, all parameters of the algorithm are estimated *on the fly* to adapt its behavior to the video content, the encoding configuration, and the target complexity over time. This method was published in IEEE Transactions on Multimedia [Jiménez-Moreno et al., 2016].

The remainder of this chapter is organized as follows. Section 5.1 gives a brief introduction. In Section 5.2, an overview of the state-of-the-art methods that address complexity control for the HEVC standard is presented along with the main contributions of our proposal. Section 5.3 provides a detailed explanation of the proposed method. In Section 5.4, the experimental results supporting the proposal are presented and discussed. Finally, Section 5.5 summarizes our conclusions.

5.1 Introduction

As HEVC is the most recent standard, it is receiving great attention in the research community [Kim et al., 2012, Zhao et al., 2013, Choi and Jang, 2012, Teng et al., 2011, Correa et al., 2011, Grellert et al., 2013, Leng et al., 2011, Shen et al., 2012, Ahn et al., 2015, Lee et al., 2015]. However, there is still a great room for the development of computationally efficient coding algorithms.

The extremely high computational complexity of the HEVC standard, due to its quadtree coding structure, makes especially relevant the design of algorithms capable of adapting its complexity to that required by specific devices and applications. Thus, the motivation of this work is to design an algorithm that is able to control the computational complexity of an HEVC video encoder and, consequently, enables more efficient implementations of the HEVC standard. Using the statistical properties of the sequences and time measures of the encoder, we can adjust the encoding process to satisfy the required target complexity. Moreover, we propose to adaptively adjust the parameters of our method, in an attempt to avoid the generalization problem associated with the use of a fixed configuration coming from an *off-line* training stage.

5.2 Review of the state of the art

The complexity reduction and complexity control problems in HEVC have typically been addressed by providing fast solutions to different sub-problems concerned with the determination of the CU depths, PU modes, TU sizes, or combinations thereof.

Relevant works exist to achieve efficient solutions for PU mode selection (e.g., [Kim et al., 2012, Zhao et al., 2013, Gweon et al., 2011, Blasi et al., 2013]) or TU size selection (e.g., [Choi and Jang, 2012] and [Teng et al., 2011]). Nevertheless, we focus here on describing the state-of-the-art methods for fast CU depth determination and the complexity control problem itself, as our goal is to design a method to control the complexity of an HEVC encoder based on fast CU depth determination.

In [Correa et al., 2011], a complexity control method for HEVC was proposed. The

method relies on the observation that in co-located regions of consecutive frames in which certain features (motion, texture, etc.) remain unaltered, the same CU depths tend to be selected as optimal. Thus, this information can be used in subsequent frames, which the authors called "constrained frames", thereby avoiding the necessity of assessing the remaining CUs. The complexity control is achieved by estimating the number of constrained frames between each pair of regularly encoded frames that is required to meet the complexity target. Furthermore, the same authors presented an extension of this work [Correa et al., 2013] with the intent of improving the previous solution for the case of fast-motion video sequences with low target complexities. Specifically, they proposed to estimate the maximum CU depth to be explored in the constrained frames based on both spatial and temporal correlations, such that spatial neighboring CTBs are also used along with the CTBs of previous frames to estimate the CU depths for the current constrained frame. In [Ukhanova et al., 2013], a rate-distortion and complexity optimization method was proposed for the selection of quantization and depth parameters. Using predictive techniques and game-theory-based methods, the maximum CU depths for certain frames are restricted to allow the encoding to be performed within a given complexity budget. In [Grellert et al., 2013], a workload management scheme was suggested. The underlying idea of this scheme is to dynamically control the complexity by estimating a complexity budget for each frame. Specifically, this method acts on different parameters of the encoder, such as the maximum CU depth assessed, the search range, etc., to achieve the complexity target. To minimize the R-D losses, a set of different encoding configurations was designed, as well as a control feedback loop to dynamically update the available computational resources.

[Ahn et al., 2015] presented a fast CU encoding scheme based on the spatio-temporal encoding parameters of HEVC. This method utilizes spatial encoding parameters such as SAO filter data to estimate the texture complexity in a CU partition. Moreover, the temporal complexity is estimated by means of temporal encoding parameters such as MVs or TU sizes. All these parameters were used to design an early CU Skip mode detection and a fast CU split decision methods. [Lee et al., 2015]

proposed three methods to save complexity: a Skip mode detection, an early CU termination, and a CU skip estimation. The first one determines for each CU whether the Skip mode should be the only mode tested. The second and third methods aim to decide if either a larger or a smaller CU size should be evaluated, respectively. The three methods are based on PDFs of the RD costs and use the Bayes' rule to make the corresponding decisions. In [Leng et al., 2011], a fast CU depth decision method was described. Based on an analysis of the CU depths selected in the last frame, the least-used CU depths are disabled in the current frame. Moreover, for every CTB, the depths of neighboring and co-located CTBs are analyzed to avoid unnecessary checking of the CU depths. The decisions at the frame level depend on certain thresholds, and the decisions at the CU level depend on the number of neighboring CTBs that fulfill certain requirements. The thresholds and the number of neighbors are determined experimentally. In [Shen et al., 2012], a fast CU depth selection method relying on a Bayesian approach was presented. The algorithm is based on the *off-line* estimation of several class-conditional PDFs that are then stored in a look-up table. Subsequently, using a Bayesian decision rule, the thresholds to determine whether to check the next CU depth are estimated (also *off-line*) for different encoding settings and sequence resolutions. [Shen et al., 2013] also presented a method that is able to constrain the CU depths by predicting the optimal depth from spatially neighboring and co-located CTBs. Moreover, three early termination methods for selecting a suitable PU partitioning were also described. [Xiong et al., 2014] presented a method based on so-called "pyramid motion divergence" for the early skipping of certain CU depths. First, the optical flow is estimated from a down-sampled original (non-encoded) frame. Then, for each CTB, the pyramid motion divergence is calculated as the variance of the optical flow of the current CTB with respect to that of the CTBs of smaller size. Finally, because CTBs with similar pyramid motion divergence values tend to use similar splittings, an algorithm based on Euclidean distances is used to select a suitable CU quadtree structure. In [Choi et al., 2011], a simple early termination method for CUs was suggested. Specifically, the authors proposed to terminate the CTB splitting process when the selected PU

mode for the current CU depth is the Skip mode. In [Tan et al., 2012], a method was presented that addresses decisions at the CU and PU levels. The PU mode decision is terminated if the R-D cost is below a certain threshold. This threshold is calculated as the average R-D cost of certain blocks previously encoded using the Skip mode. The early CU depth decision is made by comparing the R-D cost at the current depth with the sum of the best four R-D costs of the CUs of the subsequent depth. If the latter is higher, then the CTB is not split further. [Zhang et al., 2013] presented a CU depth decision method based on the depth correlation information between adjacent CTBs in the same spatial and temporal neighborhoods. The depth search range is adaptively selected for each CTB based on the information regarding the most frequently selected CU depths among spatially and temporally neighboring CTBs. Using such information for the CTB to be encoded, a similarity degree is selected, from which different depth search ranges are available.

Several previous complexity control efforts suffer from slow convergence to the target complexity (e.g., [Correa et al., 2011] and [Correa et al., 2013]). In other cases, the complexity is controlled by means of the dynamic selection of encoding configuration parameters without accounting for the potential impact of every parameter on the performance (e.g., [Grellert et al., 2013]). Other methods are unable to adapt to the video content (e.g., [Leng et al., 2011] or [Shen et al., 2012]; these are actually complexity reduction methods that rely on either fixed thresholds or statistics that are calculated *off-line*).

The main contribution of this work is the use of a CU early termination method that is based on an R-D cost analysis for the design of a complexity control method. These techniques are commonly used in complexity reduction frameworks, but they have not usually been applied to address the complexity control problem in HEVC. Our proposal relies on adaptive thresholds computed based on R-D cost statistics and actual encoding time measures to control the complexity *on the fly*. In this way, the behavior of the method adapts over time to the video content, the encoder configuration, and variations in the target complexity. Finally, our method requires only simple mathematical operations to update the parameters; in other words, there

is no additional complexity associated with the method itself.

5.3 Proposed method

This work is inspired by that presented in the previous Chapter 4 in the H.264/AVC standard. The method described in Chapter 4 proposed a hypothesis test that is used to choose either a low- or a high-complexity encoding mode and the threshold for this test is adapted with respect to the target complexity. Although the MD process in H.264 is more closely related to the PU selection task in HEVC, the method proposed in this chapter focuses on the CU depth decision, which is a more effective process from which to tackle the complexity control problem in HEVC. Consequently, a comprehensive "ad hoc" analysis was conducted to understand the encoder performance with regard to the CU depth decision method. Moreover, because the decision process must cover all available CU depths, we propose to make a split or non-split decision at every depth level that must be managed to achieve a given target complexity.

5.3.1 Overview

The proposed method establishes an early termination condition at each CU depth based on a set of dynamically adjusted thresholds. Figure 5.1 shows a flowchart that summarizes the main steps of the method. For every frame f and every CTB c, the method begins exploring all possible PU modes at the lowest CU depth level $i = 0$. If the early termination condition is satisfied, then higher depth levels are not tested, saving the corresponding computational time. By contrast, if early termination does not occur, the encoder continues evaluating the next CU depth $i = i+1$, again checking the corresponding termination condition. In summary, the goal of the proposed method is to determine a suitable number of depths to explore such that a given complexity constraint is met and the encoder does not incur significant losses in coding efficiency. To that end, the proposed method must be content-dependent, i.e., must adapt to the actual video sequence content; moreover, the bit rate and quality must be maintained near those achieved in the regular coding process.

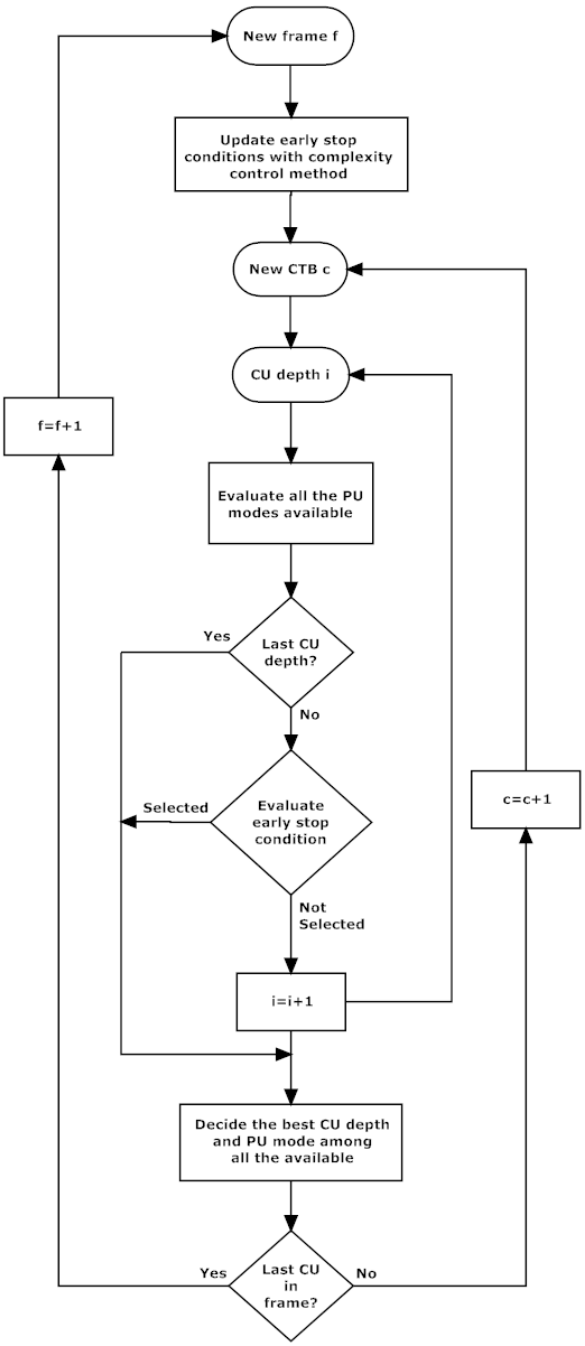


Figure 5.1: Flowchart of the proposed method for complexity control in HEVC.

The thresholds depend on the statistics of the R-D costs and the actual time encoding measures. The R-D costs (3.6) help us to select an appropriate CU depth while not incurring significant losses in coding efficiency. The time encoding measures

allow us to adjust the encoding process to satisfy the target complexity requirement in real time. In other words, by virtue of having real-time measures of the time spent in the encoding process, we are able to dynamically adjust the thresholds to achieve a suitable number of early terminations to meet the target complexity.

5.3.2 Feature selection

In this section we study whether the R-D costs are adequate variables to design a fast CU decision algorithm intended to control the complexity in HEVC. The objective of our analysis is threefold: first, to study the relationships between the CU depths and both the actual video content of the sequences and the QP; second, to check whether the R-D costs are useful for making early CU depth decisions in HEVC; and third, to determine which of the available R-D costs is most suitable for our purposes.

Relationship between CU depth and QP

First, we study the *a priori* probabilities of every CU depth in several video sequences and for different quality targets. In this manner, we intend to gain some insight into the relationships between the optimal CU depths and the contents of the sequences and the QP. For this purpose, we used the HM13.0 software[1] with the configuration file *encoder_lowdelay_P_main* (with this configuration, the maximum CU size is 64×64 pixels, corresponding to depth 0, and the minimum is 8×8 pixels, corresponding to depth 3; moreover, an IP coding pattern is used). Following the specifications given in [Bossen, 2011], a subset of the recommended test sequences was encoded with QP values of 22, 27, 32, and 37. Under these conditions, we estimated the *a priori* probabilities of every CU depth considering the complete encoded sequence. For brevity, we show in Table 5.1 the results for only three representative sequences because similar conclusions can be drawn from the others.

As can be observed, for the sequences with smooth or little movement and static regions (such as *FourPeople*), the encoder selects the lower depths with high probability for all QP values. However, the probability of selecting lower depths decreases

[1]High Efficiency Video Coding [Online]. Available: http://hevc.hhi.fraunhofer.de/.

		Depth 0	Depth 1	Depth 2	Depth 3
BasketballDrill	**QP 22**	0.17	0.33	0.25	0.19
	QP 27	0.33	0.29	0.21	0.11
	QP 32	0.44	0.27	0.18	0.05
	QP 37	0.54	0.16	0.13	0.03
BQMall	**QP 22**	0.16	0.25	0.30	0.23
	QP 27	0.26	0.27	0.25	0.15
	QP 32	0.35	0.29	0.22	0.09
	QP 37	0.44	0.29	0.17	0.05
FourPeople	**QP 22**	0.50	0.27	0.16	0.06
	QP 27	0.71	0.18	0.08	0.02
	QP 32	0.81	0.13	0.04	0.01
	QP 37	0.86	0.09	0.03	0.01

Table 5.1: *A priori* probabilities (%) of every CU depth.

notably when the sequence is more complex. Moreover, the *a priori* probability of depth 0 (or even depth 1 in *BQMall*) increases with an increasing QP. This means that a coarser coding process results in a larger number of CUs being encoded at large sizes (such as 64×64 or 32×32). In view of these results, which indicate that low CU depths tend to be optimal for several types of sequences and QP values, an early determination of the optimal CU depth favoring low depths should undoubtedly contribute to reducing the complexity of the encoding process. Moreover, it seems reasonable to design an algorithm that is able to adapt its behavior *on the fly* with respect to the content and the encoder configuration, considering that the optimal depth exhibits a strong dependence on both the particular sequence and the QP value.

Analysis of the RD costs

We study whether the R-D costs are suitable variables to make early decisions regarding the CU depth in the HEVC standard. At every CU depth, several PU modes will be evaluated by the encoder in R-D terms to select the best mode for that depth. We denote by $J_{PU=a,depth=i}$ the R-D cost associated with PU mode a at depth i. Specifically, we analyze the statistics (means and standard deviations) of the R-D costs associated with every mode a at depth i when depth i is optimal and when it

is not, i.e., $J_{PU=a,depth=i}|depth^* = i$ and $J_{PU=a,depth=i}|depth^* \neq i$, respectively.

The experimental setup was the same as for the previous analysis. In Table 5.2, we show the results for two sequences (*BasketballDrill* and *FourPeople*) at the four recommended QP values for depth 0. $J_{Merge,0}$ refers to the R-D cost associated with the Merge PU mode at depth 0, $J_{Min,0}$ refers to the minimum cost obtained after all available PU modes have been checked (depth 0), $J_{2N\times2N,0}$ refers to the cost for the 2N×2N PU mode (depth 0), and so on. $\mu|d^* = 0$ and $\sigma|d^* = 0$ denote the mean and standard deviation, respectively, when the optimal depth is 0, whereas $\mu|d^* \neq 0$ and $\sigma|d^* \neq 0$ denote the same statistics when the optimal depth is other than 0.

			$J_{Merge,0}$	$J_{2N\times2N,0}$	$J_{2N\times N,0}$	$J_{N\times2N,0}$	$J_{Min,0}$
BasketballDrill	**QP 22**	$\mu\|d^* = 0$	27242	27465	27224	27237	26986
		$\sigma\|d^* = 0$	4422	4742	4427	4464	4383
		$\mu\|d^* \neq 0$	59646	59136	57142	56345	54722
		$\sigma\|d^* \neq 0$	35805	34893	32894	31973	30841
	QP 27	$\mu\|d^* = 0$	52231	52530	52411	52396	51713
		$\sigma\|d^* = 0$	11363	11186	10967	10952	10886
		$\mu\|d^* \neq 0$	138970	134770	129340	127230	123050
		$\sigma\|d^* \neq 0$	87177	82628	77502	75642	72824
	QP 32	$\mu\|d^* = 0$	98320	99380	99605	99376	96781
		$\sigma\|d^* = 0$	38847	36279	35970	35895	35421
		$\mu\|d^* \neq 0$	295040	281590	268250	263810	253540
		$\sigma\|d^* \neq 0$	180690	169120	156780	153640	147240
	QP 37	$\mu\|d^* = 0$	205820	208940	209950	208570	199530
		$\sigma\|d^* = 0$	137410	129050	129360	128330	127230
		$\mu\|d^* \neq 0$	568130	541880	515360	506030	482480
		$\sigma\|d^* \neq 0$	338160	318710	295790	289790	277560
FourPeople	**QP 22**	$\mu\|d^* = 0$	14940	15106	15088	15071	14911
		$\sigma\|d^* = 0$	6668	6651	6654	6637	6644
		$\mu\|d^* \neq 0$	29571	29466	28974	28959	28489
		$\sigma\|d^* \neq 0$	15777	15674	14983	14953	14426
	QP 27	$\mu\|d^* = 0$	25449	26117	26025	25970	25344
		$\sigma\|d^* = 0$	15031	14940	14952	14917	14925
		$\mu\|d^* \neq 0$	62849	62140	60130	60165	58397
		$\sigma\|d^* \neq 0$	39621	39044	36588	36405	34723
	QP 32	$\mu\|d^* = 0$	47860	50539	50227	50011	47545
		$\sigma\|d^* = 0$	33492	33115	33232	33151	33014
		$\mu\|d^* \neq 0$	134010	131760	126510	126930	122180
		$\sigma\|d^* \neq 0$	89218	86372	80503	79659	75947
	QP 37	$\mu\|d^* = 0$	100610	109720	109340	108720	99958
		$\sigma\|d^* = 0$	76376	75367	75805	75896	75265
		$\mu\|d^* \neq 0$	281280	278290	267870	268910	256700
		$\sigma\|d^* \neq 0$	170070	166500	156110	154610	146940

Table 5.2: Means and standard deviations of the R-D costs associated with every PU mode when depth 0 is optimal and when it is not.

To visually check whether the two conditional PDFs of the R-D costs (when depth i is optimal and when it is not) are truly separable, we estimated the following PDFs:

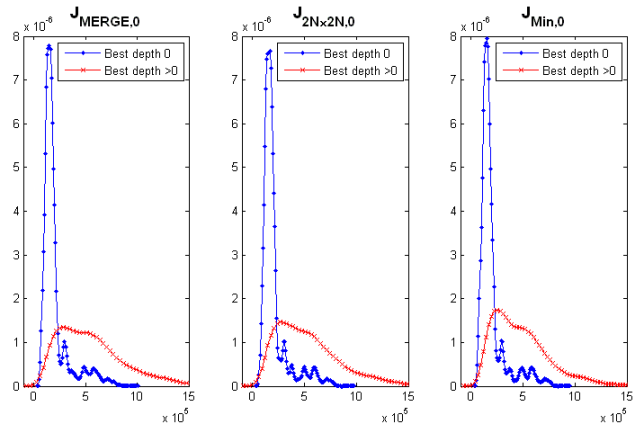
$$\begin{aligned} &\Pr_J\left(J_{PU=a,depth=i}|depth^* = i\right) \\ &\Pr_J\left(J_{PU=a,depth=i}|depth^* \neq i\right), \end{aligned} \tag{5.1}$$

which represent the probabilities of a certain cost $J_{PU=a,depth=i}$ when the optimal CU depth is i and when it is not, respectively. Figure 5.2 shows the PDFs thus obtained for depth 0 (for higher depths, the results are similar). The sequences *BasketballDrill* at QP 37 and *FourPeople* at QP 22 are used for illustration. From this figure, we gain some insight into the shape and actual separability of these PDFs.

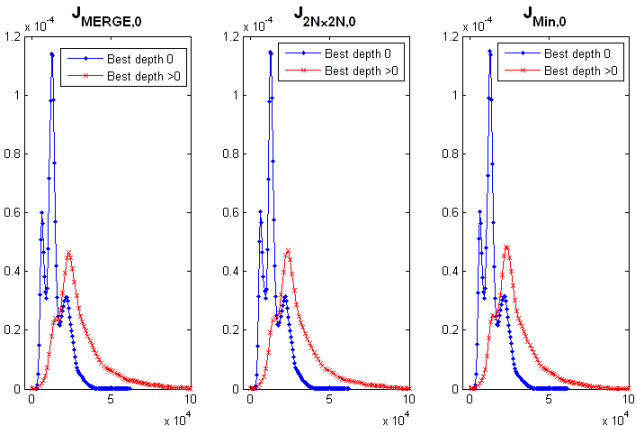
The results presented in both Table 5.2 and Figure 5.2 show that the statistics of the two compared PDFs (those corresponding to when a certain depth is optimal or not) are significantly different and reasonably separable for all analyzed R-D costs and that all analyzed $J_{PU=a,depth=i}$ costs exhibit very similar behavior. In particular, we draw two conclusions: 1) the CU depth decision problem can be addressed based on the updated statistical information regarding these R-D costs, and 2) any of the considered R-D costs would be suitable for this purpose.

Selection of the best R-D cost

From the results in the previous section it can be seen that there are relevant differences between the PDFs corresponding to the scenarios in which a certain depth is optimal or is not optimal (i.e., between whether a split decision is made at a certain depth). Moreover, each pair of PDFs of the form given in (5.1) can be separated with the establishment of a proper threshold. However, because of these differences are consistent for different R-D costs (i.e., similar behavior is observed in Figure 5.2 for $J_{Merge,0}$, $J_{2N\times 2N,0}$, and $J_{Min,0}$), it is difficult to conclude which of these R-D costs could be the optimal in the CU depth decision problem. For this reason, to find the best possible design for our complexity control proposal, we numerically measure the encoding results that are obtained using these three J costs. Specifically, we imple-



(a) *BasketballDrill* at QP 37



(b) *FourPeople* at QP 22

Figure 5.2: An illustration of the PDFs for several R-D costs at CU depth 0 for two sequences, *BasketballDrill* and *FourPeople*.

ment three versions of our proposal, each one using one R-D cost among $J_{Merge,i}$, $J_{2N\times 2N,i}$, and $J_{Min,i}$. The experimental setup is maintained as previously. To evaluate the coding performance we measure the bit rate increment $BDBR(\%)$ and the PSNR loss $BDPSNR(dB)$ following the recommendations in [G.Bjontegaard, 2001]. The time saving $TS(\%)$ is calculated as follows:

$$TS = \frac{Time(HM13.0) - Time(Proposed)}{Time(HM13.0)} \times 100. \tag{5.2}$$

As it can be seen in Table 5.3 for some selected examples (the results are obtained

		$BDBR$	$BDPSNR$	TS
Cactus	$J_{Merge,i}$	3.04	0.06	48.02
	$J_{2N\times2N,i}$	1.87	0.03	47.34
	$J_{Min,i}$	1.88	0.04	46.49
Kristen	$J_{Merge,i}$	0.66	0.01	45.96
	$J_{2N\times2N,i}$	0.30	0.01	45.83
	$J_{Min,i}$	0.49	0.01	45.56
Park	$J_{Merge,i}$	2.94	0.09	35.32
	$J_{2N\times2N,i}$	1.75	0.05	33.80
	$J_{Min,i}$	1.75	0.05	33.67
Vidyo4	$J_{Merge,i}$	1.78	0.05	47.85
	$J_{2N\times2N,i}$	1.33	0.04	48.61
	$J_{Min,i}$	1.44	0.04	47.61

Table 5.3: $BDBR$, $BDPSNR$, and TS for three versions of our proposal using $J_{Merge,i}$, $J_{2N\times2N,i}$, or $J_{Min,i}$ R-D costs, respectively.

for the sequences *Cactus*, *KristenAndSara*, *ParkScene*, and *Vidyo4* averaging the obtained values for all the considered QPs), slightly better results are achieved with $J_{2N\times2N,i}$. So, we decided to use this cost as input feature.

5.3.3 Designing the early termination conditions

To control the complexity of the encoding process, the thresholds are updated *on the fly* such that the fast CU depth decision process is actually content-dependent.

To simplify the process, we rely on only one PDF to make our decision; i.e., instead of seeking an optimal threshold while accounting for both, the conditional PDF given $depth^* = i$ and the conditional PDF given $depth^* \neq i$, we make our decision at every depth level by considering only $(\Pr_J(J_{PU=a,depth=i}|depth^* = i))$, as we successfully proposed in [Martinez-Enriquez et al., 2010] for the H.264/AVC framework. Though, generally, taking into account both PDFs would result in more accurate decisions, we checked the two proposals and we obtained slightly better results with the thresholds built from just one PDF. This is probably due to a convergence issue. Working with two PDFs, the number of samples required to obtain an accurate estimation of both PDFs for every CU depth will be higher than working with just one PDF and, therefore, the encoding time without applying the early decisions will be also higher.

Thus, for every CU depth i, we set a threshold that directly depends on the mean and standard deviation of the PDF ($\Pr_J\left(J_{PU=a,depth=i}|depth^*=i\right)$), i.e.,

$$TH_{depth=i} = \mu_{J_{2N\times 2N,i}} + \left(n_{depth=i} \times \sigma_{J_{2N\times 2N,i}}\right), \tag{5.3}$$

where $TH_{depth=i}$ is the threshold for depth level i; $\mu_{J_{2N\times 2N,i}}$ and $\sigma_{J_{2N\times 2N,i}}$ are the mean and standard deviation, respectively, of $\Pr_J\left(J_{2N\times 2N,i}|depth^*=i\right)$; and $n_{depth=i}$ is the control parameter, which allows us to adapt the threshold for every depth i. Defining the threshold in this manner provides two essential advantages with respect to other proposals in the state-of-the-art: 1) because the mean and standard deviation are updated on a CU-by-CU basis (see details below), the thresholds are content-adaptive, and 2) the $n_{depth=i}$ parameter allows us to set more or less demanding thresholds depending on the target complexity.

For each CTB at CU depth level i, if the actual cost $J_{2N\times 2N,i}$ is below the threshold, then the early termination condition is satisfied and the process of encoding that CTB is stopped. Otherwise, if $J_{2N\times 2N,i}$ is above the threshold, the encoding process for that CTB continues to explore higher CU depth levels. In other words, an early termination actually occurs at a particular depth level i if the likelihood of the actual cost $J_{2N\times 2N,i}$ coming from the conditional PDF given that $depth^*=i$ is sufficiently high (relative to the threshold).

Finally, the parameter $n_{depth=i}$ establishes a balance in the *complexity-coding efficiency* trade-off for a given sequence, encoding configuration, and complexity target. In particular, the control of this trade-off allows us to manage how often an early termination at depth i actually occurs and, consequently, allows us to dynamically control the complexity. Further details concerning how to manage this parameter are given in Section 5.3.4.

5.3.3.1 On the fly estimation of the statistics

The fact that the thresholds that define the early termination conditions are content-adaptive is one of the main contributions of this proposal with respect to other

state-of-the-art approaches. In this subsection, we describe how the PDF parameters are estimated *on the fly* to adapt the threshold to the content throughout the video sequence.

We model the PDFs based on their means and variances. Thus, following the same reasoning presented in Chapter 4, we apply a simple procedure for updating these two parameters on a CU-by-CU basis based on an exponential moving average:

$$\hat{\mu}_{J_{2N\times 2N,i}}(t) = \alpha\hat{\mu}_{J_{2N\times 2N,i}}(t-1) + (1-\alpha)J_{2N\times 2N,i}(t), \tag{5.4}$$

$$\hat{\sigma}^2_{J_{2N\times 2N,i}}(t) = \alpha\hat{\sigma}^2_{J_{2N\times 2N,i}}(t-1) + (1-\alpha)(J_{2N\times 2N,i}(t) - \hat{\mu}_{J_{2N\times 2N,i}}(t))^2, \tag{5.5}$$

where t is an index associated with the number of times that depth level i is selected as optimal; $\hat{\mu}_{J_{2N\times 2N,i}}(t-1)$ and $\hat{\mu}_{J_{2N\times 2N,i}}(t)$ are the estimated means at times $(t-1)$ and t, respectively (the variances are $\hat{\sigma}^2_{J_{2N\times 2N,i}}(t-1)$ and $\hat{\sigma}^2_{J_{2N\times 2N,i}}(t)$, following the same notation); $J_{2N\times 2N,i}(t)$ is the R-D cost at time t; and α is the parameter defining the forgetting factor of the exponential averaging process. Specifically, α was set to 0.95 in our experiments.

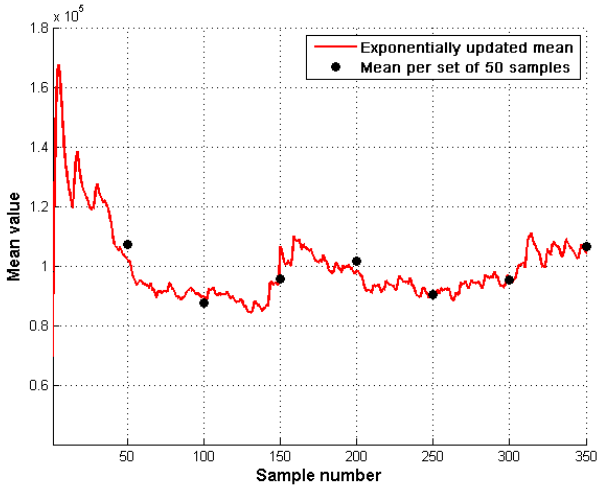


Figure 5.3: Illustration of the estimation of the exponential average $\hat{\mu}_{J_{2N\times 2N,i}}$ over time for *BasketballDrill* at QP 32.

Figure 5.3 provides an example illustrating the behavior of the exponential average. Specifically, we show the estimated means ($\hat{\mu}_{J_{2N\times 2N,i}}$) for 350 consecutive samples. Furthermore, the expected mean values considering blocks of 50 samples are also shown as control points representing the trend that the average estimator should follow. As it was expected from the results in the previous chapter, the exponential average produces a good approximation. The behavior of the estimation of the variances is nearly identical.

5.3.4 Controlling the complexity

As stated previously, $n_{depth=i}$ controls the number of early terminations occurring at depth level i. In particular, according to (5.3), a higher $n_{depth=i}$ results in a higher threshold and a greater likelihood of early termination (because the likelihood of obtaining a $J_{2N\times 2N,i}$ cost that is lower than $TH_{depth=i}$ increases). Therefore, by modifying $n_{depth=i}$, we will be able to manage the computational complexity to reach a certain target. Hereafter, we will use the encoding time as a practical measure of the computational complexity.

Following a strategy similar to that proposed for H.264/AVC (Chapter 4), we rely on a simple feedback algorithm for updating $n_{depth=i}$:

$$n_{depth=i}^{f=j} = n_{depth=i}^{f=j-1} + \lambda_{depth=i}\left(time^{f=j-1} - time_{target}\right), \tag{5.6}$$

where $f = j$ and $f = j-1$ represent frame numbers j and $j-1$, respectively; thus, $n_{depth=i}^{f=j}$ varies from its previous value (that of frame $j-1$) according to the deviation of the actual encoding time for frame $j-1$ with respect to the target, i.e., $\left(time^{f=j-1} - time_{target}\right)$. Furthermore, the parameter $\lambda_{depth=i}$ controls the trade-off between the speed and accuracy of the algorithm.

In short, $n_{depth=i}$ is updated *on the fly* on a frame-by-frame basis to satisfy the complexity constraint. For example, if $time^{f=j-1}$ is higher than $time_{target}$, more than the targeted resources were allocated to frame $j-1$, and the threshold must be increased to induce more early terminations. This occurs because $time^{f=j-1}$ –

$time_{target}$ becomes positive.

It should be noted that the parameter $\lambda_{depth=i}$ must depend on the depth level i. In doing so, we aim to produce smoother transitions of $n_{depth=i}$ when i is low, which is desirable because if we make an incorrect early termination decision at a low depth, we will incur a significant loss in coding efficiency. Thus, we allow only smooth transitions of $n_{depth=i}$ at low depths and relax this constraint as the depth increases. In particular, $\lambda_{depth=i}$ is computed as follows:

$$\lambda_{depth=i} = \lambda_0 \times (time_{depth=i}/time_{CTB}) \,, \tag{5.7}$$

where λ_0 is an initial value experimentally set to 0.2, $time_{depth=i}$ is the time required to encode a CTB with an early termination at CU depth i, and $time_{CTB}$ is the time required to encode a CTB when all available depths are explored. It is readily apparent that $time_{CTB}$ is the maximum value of $time_{depth=i}$ and that $time_{depth=i+1} > time_{depth=i}$. Therefore, when depth i is lower, $\lambda_{depth=i}$ will also be lower. It should be noted that $time_{depth=i}$ and $time_{CTB}$ are computed on a frame-by-frame basis as the average time spent encoding a CTB with an early termination at CU depth i and the average time devoted to a CTB when all depths are explored, respectively.

The target complexity can be specified by the user or by the application running the video encoder, and it depends on the resources available for a specific device and the time that the user/application allocates for the encoding process. In the proposed method, the target complexity is specified as a percentage with respect to the complexity of the full-search encoding process, $TC(\%)$, and the method transforms this percentage into a target encoding time $time_{target}$ to be used in (5.6). Specifically, if we know the time that is required to encode a CTB using the full-search approach ($time_{CTB}$) and the total number of CTBs per frame M, then we can easily obtain $time_{target}$ as follows:

$$time_{target} = time_{CTB} \times M \times (TC/100) \,. \tag{5.8}$$

5.3.5 Summary of the algorithm

The complete method is summarized in Algorithm 2.

Algorithm 2 Proposed coding process.

Require: F: number of frames to be encoded
Require: M: number of CTBs per frame
Require: D: number of available CU depths in a CTB

for $\forall f \in F$ **do**
  Retrieve the time required to encode the previous frame $time^{f=f-1}$
  Obtain average values $time_{CTB}$ and $time_{depth=d}$
  Calculate $time_{target}$ to encode frame f (5.8)
  Calculate $\lambda_{depth=d}$ (5.7) and $n_{depth=i}$ (5.6)
  for $\forall m \in M$ **do**
    for $\forall d \in D$ **do**
      Evaluate all PU partitioning modes at depth d
      Calculate $\hat{\mu}_{J_{2N\times 2N,d}}$ (5.4) and $\hat{\sigma}^2_{J_{2N\times 2N,d}}$ (5.5)
      Obtain $TH_{depth=d}$ (5.3)
      if $J_{2N\times 2N,d} < TH_{depth=d}$ **then**
        Go to 15
      end if
    end for
    Determine the best coding options among all evaluated CU depths and PU partitioning modes
  end for
end for

5.4 Experiments and results

5.4.1 Experimental setup

To assess the performance of the proposed method, it was implemented in the HM13.0 software. The test conditions were chosen following the recommendations given in [Bossen, 2011], and the configuration file was "encoder_lowdelay_P_main". A comprehensive set of sequences at several resolutions and covering a variety of video contents (motion, textures, etc.) was used (see Table 5.4).

5.4.2 Achieving a target complexity

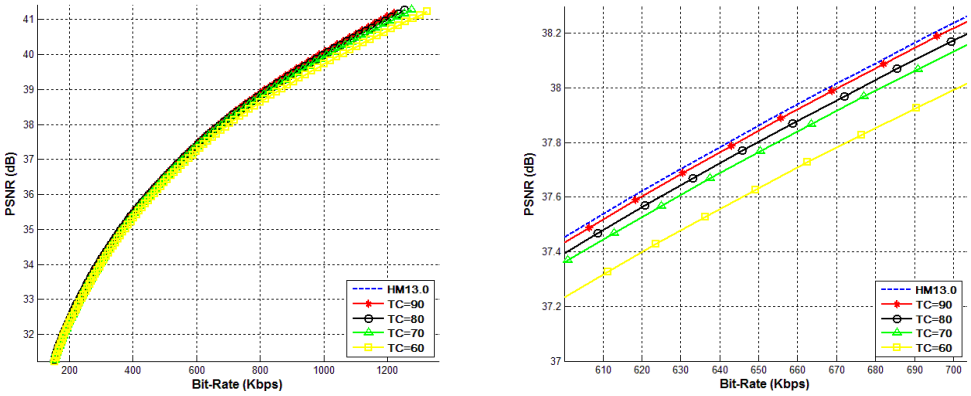
To measure the ability of our proposed method to achieve different target complexities and the possible losses in coding efficiency, we use the measures of TS, $BDBR(\%)$, and $BDPSNR(dB)$ defined in Section 5.3.2. Moreover, we evaluated our proposal for four different target complexities, $TC(\%) = \{90, 80, 70, 60\}$.

TC	$TC = 90\%$			$TC = 80\%$			$TC = 70\%$			$TC = 60\%$		
Sequence	$BDBR$ (%)	$BDPSNR$ (dB)	TS (%)	$BDBR$ (%)	$BDPSNR$ (dB)	TS (%)	$BDBR$ (%)	$BDPSNR$ (dB)	TS (%)	$BDBR$ (%)	$BDPSNR$ (dB)	TS (%)
BasketballPass (416×240)	0.42	0.02	13.68	1.11	0.06	20.44	2.10	0.11	27.82	4.84	0.25	35.15
BlowingBubbles (416×240)	0.90	0.02	12.62	2.26	0.08	20.43	5.74	0.24	32.95	11.79	0.47	43.75
BQSquare (416×240)	0.54	0.01	11.56	1.40	0.05	17.75	3.74	0.15	26.50	12.74	0.58	39.14
RaceHorses (416×240)	0.12	0.01	5.60	0.92	0.05	13.77	3.82	0.21	24.17	15.42	0.78	38.92
BasketballDrill (832×480)	0.67	0.02	9.57	0.79	0.02	17.06	1.23	0.05	28.44	5.19	0.24	39.76
BasketballDrillText (832×480)	0.48	0.02	8.52	0.75	0.03	17.28	1.11	0.05	27.75	6.35	0.30	41.12
BQMall (832×480)	0.31	0.01	12.42	0.71	0.03	20.78	2.42	0.11	31.44	12.20	0.49	46.59
PartyScene (832×480)	1.13	0.05	9.97	3.62	0.17	21.18	11.40	0.55	37.91	25.05	1.02	52.67
ChinaSpeed (1024×768)	0.17	0.01	12.73	0.89	0.07	21.15	4.03	0.29	30.58	26.67	1.38	42.66
FourPeople (1280×720)	−0.01	0.00	12.55	0.16	0.01	20.93	0.35	0.01	31.57	0.66	0.02	44.46
Johnny (1280×720)	−0.04	0.00	9.74	0.08	0.00	19.35	−0.06	0.00	28.84	−0.10	0.00	39.44
KristenAndSara (1280×720)	0.12	0.00	9.46	−0.18	0.00	18.39	0.07	0.00	28.08	0.12	0.00	39.08
SlideEditing (1280×720)	0.31	0.04	8.78	0.23	0.02	17.66	0.93	0.13	28.96	0.48	0.07	42.49
SlideShow (1280×720)	−0.22	−0.01	7.06	0.50	0.04	16.58	1.61	0.14	25.16	6.97	0.54	35.56
Vidyo1 (1280×720)	0.12	0.00	9.75	0.24	0.01	18.61	0.14	0.00	28.80	0.26	0.01	39.48
Vidyo3 (1280×720)	−0.05	0.00	7.16	−0.17	0.00	15.93	−0.08	0.00	26.74	0.46	0.01	41.46
Vidyo4 (1280×720)	−0.02	0.00	10.13	0.01	0.00	18.07	0.29	0.01	29.32	0.87	0.04	44.83
BasketballDrive (1920×1080)	0.07	0.00	9.30	0.32	0.01	17.10	0.83	0.02	26.51	1.72	0.05	38.21
BQTerrace (1920×1080)	0.06	0.00	9.98	0.59	0.02	20.60	1.77	0.06	33.65	7.92	0.25	54.10
Cactus (1920×1080)	0.10	0.00	7.19	0.14	0.01	14.91	0.74	0.03	25.22	2.42	0.08	39.58
Kimono (1920×1080)	0.04	0.00	8.14	0.29	0.01	19.69	0.72	0.03	30.25	1.07	0.04	42.06
ParkScene (1920×1080)	0.03	0.00	9.65	0.57	0.03	19.51	1.57	0.07	31.14	7.28	0.31	48.62
Average	**0.23**	**0.01**	**9.79**	**0.69**	**0.03**	**18.50**	**2.02**	**0.10**	**29.17**	**6.83**	**0.31**	**42.23**

Table 5.4: Performance evaluation of the proposed complexity control method in HEVC for different target complexities.

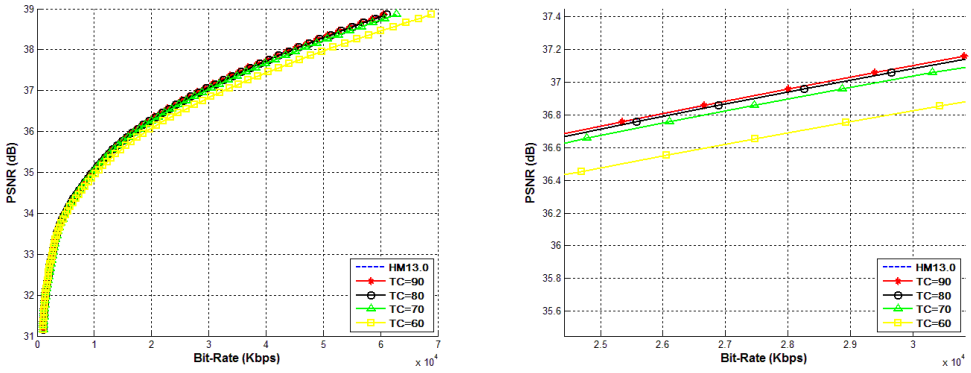
Table 5.4 shows the results for all considered target complexities in terms of $BDBR$, $BDPSNR$, and TS, averaged over the four QP values recommended in [Bossen, 2011] (QP values of 22, 27, 32, and 37). From these results, we can conclude that the proposed method achieves notable accuracy for all the considered target complexities. In particular, the mean values of TS obtained when considering all video sequences show a very small deviation from the target values (the highest deviation is 2.23%, for TC=60%). Moreover, the complexity savings are obtained in exchange for very limited losses in coding performance. The mean values of $BDBR$ are 0.23%, 0.69% and 2.2% for TCs of 90, 80, and 70%, respectively. However, when TC is 60%, the average $BDBR$ reaches 6.8%; this is expected because the proposed method acts only on the CU depth, and low target complexities necessarily involve large CU sizes and, therefore, a coarser encoding process.

Selected numerical results in Table 5.4 are further illustrated in Figures 5.4 and 5.5 to demonstrate the R-D performance of the proposed method for different target complexities. On the left-hand side of each figure, the complete R-D curves are shown, whereas on the right-hand side, we have zoomed in on the curves to show a segment in greater detail. As can be observed, the conclusions do not change with respect to those already drawn from Table 5.4.



(a) R-D performance. (b) Zoomed view of the Figure on the left.

Figure 5.4: R-D performance for the 4 considered target complexities for the sequence *BasketballPass*.



(a) R-D performance.

(b) Zoomed view of the Figure on the left.

Figure 5.5: R-D performance for the 4 considered target complexities for the sequence *BQTerrace*.

5.4.3 Comparison with a state-of-the-art complexity control method

In this subsection, we compare the proposed method with a state-of-the-art method [Correa et al., 2013]. The configuration file and QPs used to perform this comparison were the same as those used in the previous experiment, and the focus of the comparison is placed on target complexities of $TC = \{80, 70\}$ because the method described in [Correa et al., 2013] is not able to achieve $TC = 60\%$. The obtained results are shown in Table 5.5 where the values of $BDBR$, $BDPSNR$, and TS are averaged over the four QP values.

	Proposed method						[Correa et al., 2013]					
TC	$TC = 80\%$			$TC = 70\%$			$TC = 80\%$			$TC = 70\%$		
Sequence	$BDBR$ (%)	$BDPSNR$ (dB)	TS (%)	$BDBR$ (%)	$BDPSNR$ (dB)	TS (%)	$BDBR$ (%)	$BDPSNR$ (dB)	TS (%)	$BDBR$ (%)	$BDPSNR$ (dB)	TS (%)
BasketballPass (416×240)	1.11	0.06	20.44	2.10	0.11	27.82	6.27	0.32	17.30	7.67	0.37	21.39
BlowingBubbles (416×240)	2.26	0.08	20.43	5.74	0.24	32.95	2.69	0.10	16.88	3.80	0.14	21.11
BQSquare (416×240)	1.40	0.05	17.75	3.74	0.15	26.50	3.09	0.12	13.67	5.79	0.19	21.73
RaceHorses (416×240)	0.92	0.05	13.77	3.82	0.21	24.17	5.30	0.29	15.72	7.41	0.39	20.22
BasketballDrill (832×480)	0.79	0.02	17.06	1.23	0.05	28.44	4.45	0.18	19.52	6.99	0.27	25.60
BasketballDrillText (832×480)	0.75	0.03	17.28	1.11	0.05	27.75	4.17	0.19	18.34	6.54	0.28	25.51
BQMall (832×480)	0.71	0.03	20.78	2.42	0.11	31.44	4.62	0.19	19.29	7.78	0.32	23.26
PartyScene (832×480)	3.62	0.17	21.18	11.40	0.55	37.91	4.18	0.18	18.78	6.27	0.27	24.41
ChinaSpeed (1024×768)	0.89	0.07	21.15	4.03	0.29	30.58	5.76	0.34	19.66	9.11	0.50	27.77
FourPeople (1280×720)	0.16	0.01	20.93	0.35	0.01	31.57	1.73	0.05	22.53	3.32	0.08	31.15
Johnny (1280×720)	0.08	0.00	19.35	−0.06	0.00	28.84	1.82	0.02	19.98	4.01	0.05	21.55
KristenAndSara (1280×720)	−0.18	0.00	18.39	0.07	0.00	28.08	1.55	0.04	21.49	2.09	0.04	34.63
SlideEditing (1280×720)	0.23	0.02	17.66	0.93	0.13	28.96	5.14	0.83	16.99	8.76	1.25	25.64
SlideShow (1280×720)	0.50	0.04	16.58	1.61	0.14	25.16	5.72	0.42	25.42	6.68	0.47	28.53
Vidyo1 (1280×720)	0.24	0.01	18.61	0.14	0.00	28.80	1.74	0.04	20.58	2.33	0.04	31.80
Vidyo3 (1280×720)	−0.17	0.00	15.93	−0.08	0.00	26.74	0.96	0.03	18.55	2.04	0.05	30.92
Vidyo4 (1280×720)	0.01	0.00	18.07	0.29	0.01	29.32	1.09	0.03	17.80	6.10	0.10	25.39
BasketballDrive (1920×1080)	0.32	0.01	17.10	0.83	0.02	26.51	2.30	0.04	20.19	5.00	0.08	29.98
BQTerrace (1920×1080)	0.59	0.02	20.60	1.77	0.06	33.65	1.36	0.03	18.48	2.84	0.05	25.58
Cactus (1920×1080)	0.14	0.01	14.91	0.74	0.03	25.22	1.85	0.04	21.57	3.20	0.06	25.95
Kimono (1920×1080)	0.29	0.01	19.69	0.72	0.03	30.25	0.29	0.01	22.75	0.56	0.01	29.96
ParkScene (1920×1080)	0.57	0.03	19.51	1.57	0.07	31.14	2.27	0.08	21.60	3.38	0.11	27.24
Average	**0.69**	**0.03**	**18.50**	**2.02**	**0.10**	**29.17**	**3.10**	**0.16**	**19.41**	**5.07**	**0.23**	**26.33**

Table 5.5: Performance evaluation of the proposed complexity control method in HEVC in comparison with [Correa et al., 2013].

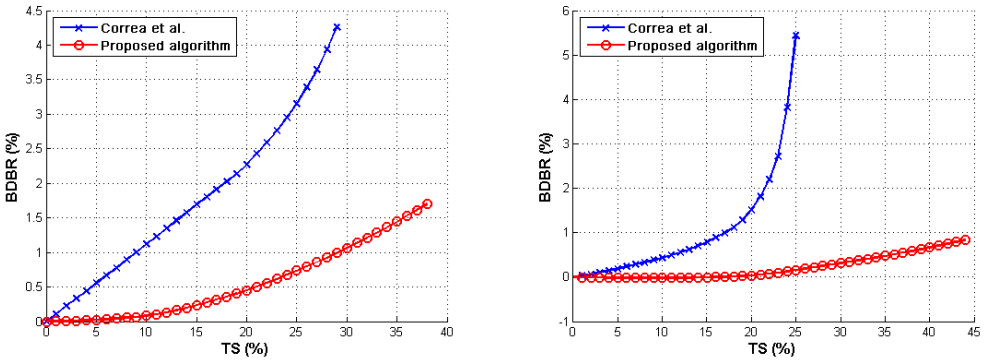
As it can seen in this table, the method presented in [Correa et al., 2013] is not as accurate as the proposed method in terms of achieving the target complexity for low TC values. In particular, for $TC = 80\%$, the deviation from the target is less than 2% in both cases; for $TC = 70\%$, the deviation of the proposed method remains below 2% whereas in the case of [Correa et al., 2013] it increases to 3.6%. Furthermore, for $TC = 70\%$ and certain specific sequences, such as *Johnny*, *BasketballPass*, and *BlowingBubbles*, [Correa et al., 2013] is not able to achieve the required TC; in fact, for these sequences, the proposed method yields significantly higher time savings (28.84, 27.82, and 32.95 %, respectively) compared with those of [Correa et al., 2013] (21.55, 21.39, and 21.11%, respectively).

Regarding losses in coding efficiency, the average $BDBR$ of [Correa et al., 2013] is higher than that of the proposed method; specifically, the results of [Correa et al., 2013] are approximately 3% worse than those of the proposed method.

As can be inferred from the results given in Table 5.5, the method presented in [Correa et al., 2013] experiences greater difficulty when attempting to reach lower target complexities. In fact, no comparison was performed for $TC = 60\%$ because of the difficulty of achieving higher TSs. This difficulty can be attributed to the fact that this method is not able to reach the high number of constrained frames required to achieve low target complexities. Moreover, if a sequence exhibits high motion content, then the optimal CU depths that are stored to be used in the constrained frames will probably be high depths, and consequently, the potential complexity savings are limited. By contrast, the proposed method does not suffer from this problem because it can always set higher thresholds to reach any required target complexity.

In Figure 5.6, we present $BDBR$ as a function of TS for the two algorithms. Specifically, we show the results obtained for the two representative sequences *BasketballDrive* (Figure 5.6(a)) and *Vidyo4* (Figure 5.6(b)). As it can be seen, the method of [Correa et al., 2013] is not able to achieve TSs greater than 30%, whereas the proposed method can produce TSs of nearly 40%. In terms of $BDBR$, our proposal is able to achieve the same target complexities in exchange for notably smaller

losses in coding efficiency, as is evident in Figure 5.6 for both sequences.



(a) *BasketballDrive*. (b) *Vidyo4*.

Figure 5.6: Performance evaluation of the proposed complexity control algorithm in HEVC in comparison with [Correa et al., 2013]. The bit rate increment is plotted as a function of the computational time savings.

5.4.4 Comparison with a state-of-the-art complexity reduction method

In this subsection, we compare our proposal with a state-of-the-art method of complexity reduction [Zhang et al., 2013], which strives to reduce the complexity of the HEVC encoder but is not capable of meeting a specified complexity target. The configuration file and QPs used to conduct this comparison were the same as those used in the previous experiments. The obtained $BDBR$, $BDPSNR$, and TS values, averaged over the four QP values, are shown in Table 5.6. To fairly compare both methods, we provide in Table 5.6 the results of the proposed method that are most similar to those of [Zhang et al., 2013] in terms of TS.

Several interesting conclusions can be extracted from this experiment. First, the method presented in [Zhang et al., 2013] is not a complexity control method and, consequently, provides very different results in terms of TS depending on the content of the video sequence (e.g., in *KristenAndSara*, the TS is 54%, whereas that in *BlowingBubbles* is 8%). Therefore, it does not provide a solution that is capable of

	Proposed method			[Zhang et al., 2013]		
Sequence	*BDBR* (%)	*BDPSNR* (*dB*)	*TS* (%)	*BDBR* (%)	*BDPSNR* (*dB*)	*TS* (%)
BasketballPass (416×240)	0.42	0.02	13.68	0.48	0.01	10.06
BlowingBubbles (416×240)	0.90	0.02	12.62	0.03	0.00	8.49
BQSquare (416×240)	0.54	0.01	11.56	0.12	0.00	8.78
RaceHorses (416×240)	0.12	0.01	5.60	0.07	0.00	7.78
BasketballDrill (832×480)	0.79	0.02	17.06	0.97	0.02	17.96
BasketballDrillText (832×480)	0.75	0.03	17.28	1.21	0.02	18.11
BQMall (832×480)	0.71	0.03	20.78	0.56	0.01	17.12
PartyScene (832×480)	1.13	0.05	9.97	0.13	0.00	12.66
ChinaSpeed (1024×768)	0.89	0.07	21.15	0.18	0.01	19.68
FourPeople (1280×720)	0.66	0.02	44.46	7.86	0.11	46.58
Johnny (1280×720)	−0.10	0.00	39.44	4.20	0.05	46.96
KristenAndSara (1280×720)	0.12	0.00	39.08	14.28	0.25	54.15
SlideEditing (1280×720)	0.48	0.07	42.49	53.95	2.66	64.81
SlideShow (1280×720)	6.97	0.54	35.56	42.66	2.21	51.30
Vidyo1 (1280×720)	0.26	0.01	39.48	9.97	0.17	49.21
Vidyo3 (1280×720)	0.46	0.01	41.46	3.05	0.04	42.94
Vidyo4 (1280×720)	0.87	0.04	44.83	6.55	0.07	42.78
BasketballDrive (1920×1080)	0.83	0.02	26.51	0.97	0.01	26.79
BQTerrace (1920×1080)	1.77	0.06	33.65	0.95	0.01	28.76
Cactus (1920×1080)	0.74	0.03	25.22	0.93	0.01	29.29
Kimono (1920×1080)	0.29	0.01	19.69	0.42	0.01	24.47
ParkScene (1920×1080)	0.57	0.03	19.51	1.31	0.02	24.89
Average	**0.91**	**0.05**	**26.41**	**6.85**	**0.25**	**29.70**

Table 5.6: Performance evaluation of the proposed complexity control method in HEVC in comparison with [Zhang et al., 2013].

running on a fixed-resource platform, whereas our approach is perfectly suitable for this purpose, as already proved in Table 5.4.

Second, the results obtained for *SlideEditing* and *SlideShow* are particularly worthy of note. In these two sequences, the content is quite uniform until a sudden change occurs. This is difficult for the algorithm of [Zhang et al., 2013] to cope with; this algorithm suffers when a scene change occurs because only a few depths are actually allowed for such frames, incurring huge *BDBRs* (approximately doubling the original rate). By contrast, our method does not suffer from this problem because of

the adaptive statistics used to define our early termination thresholds.

Third, in general terms, the results of our proposed method are superior to those of [Zhang et al., 2013] when compared at the same level of complexity reduction (e.g., *BasketballDrillText* and *Vidyo3*). For certain sequences, the results of our method are slightly worse than those of [Zhang et al., 2013] (e.g., *RaceHorses*). However, on average, our proposal outperforms [Zhang et al., 2013], as similar complexity savings are achieved at the expense of a markedly lower $BDBR$ (0.91% vs. 6.85%).

5.4.5 Convergence properties

The ability to reach the required target complexity with a certain accuracy level and in the shortest possible time is a relevant performance indicator. In this section we prove that our method is able to adapt its behavior *on the fly* throughout the coding process to any type of content, coding configuration, or complexity target.

The convergence properties of our algorithm are reported in Tables 5.7 and 5.8. The average time actually spent on encoding the frames is denoted by $\widehat{time}^{f}$, and the average target time is denoted by $\widehat{time}_{target}$. All results are given in units of seconds. The results given in Table 5.7 were obtained using a QP value of 32, and for those given in Table 5.8 the QP value used was 27.

Table 5.7 shows the results for three different sequences (*Blowing Bubbles*, *BQMall*, and *Four People*) exhibiting very different contents, each one encoded with four target complexities, $TC(\%) = \{90, 80, 70, 60\}$ (the TC values are specified on the left-hand side of the table). It is evident that the proposed algorithm achieves an encoding time that is very similar to the target; specifically, the highest deviation from the target is only 0.63 seconds. Thus, we can conclude that our proposed algorithm achieves very good accuracy.

In Table 5.8, we present several results that illustrate the behavior of the proposed method when the target complexity varies throughout the coding process. The average values of $\widehat{time}^{f}$ and $\widehat{time}_{target}$ were obtained for the same three sequences in two different cases: first, with $TC = 60\%$ from frames 0 to 49 and $TC = 80\%$ from frames 50 to 100, and second, with $TC = 90\%$ for the first fifty frames and $TC = 70\%$ for

		BlowingBubbles (416×240)	BQMall (832×480)	FourPeople (1280×720)
90	$\widehat{time}_{target}$	6.48	22.41	30.50
	$\widehat{time}^{f}$	5.85	22.09	30.20
80	$\widehat{time}_{target}$	5.13	20.28	28.35
	$\widehat{time}^{f}$	4.90	20.11	28.04
70	$\widehat{time}_{target}$	3.85	16.51	24.49
	$\widehat{time}^{f}$	3.78	16.44	24.32
60	$\widehat{time}_{target}$	3.46	12.88	19.95
	$\widehat{time}^{f}$	3.48	12.98	19.86

Table 5.7: Convergence performance evaluation of the proposed complexity control algorithm in HEVC. Evaluation with a fixed complexity target value.

the remaining frames (these sets of frames are represented in the table as "Fs 1 - 49" and "Fs 50 - 100", respectively). As observed from these examples, when a change occurs in the target complexity, the proposed algorithm is able to adapt its behavior to reach the new target complexity.

		BlowingBubbles (416×240)		BQMall (832×480)		FourPeople (1280×720)	
		Fs 1 - 49	Fs 50 - 100	Fs 1 - 49	Fs 50 - 100	Fs 1 - 49	Fs 50 - 100
60 - 80	$\widehat{time}_{target}$	4.24	5.13	14.89	24.59	21.72	32.37
	$\widehat{time}^{f}$	4.81	5.57	15.41	23.96	21.70	31.97
90 - 70	$\widehat{time}_{target}$	7.22	6.22	23.60	23.28	32.09	28.13
	$\widehat{time}^{f}$	7.24	6.04	23.64	23.25	31.50	28.51

Table 5.8: Convergence performance evaluation of the proposed complexity control algorithm in HEVC. Evaluation with a variable complexity target value.

5.5 Conclusions

In this chapter, we have proposed a complexity control method for the HEVC standard. The proposed method is based on a set of early termination conditions, one at each CU depth level, that rely on a set of thresholds that are adjusted dynamically. These thresholds are based on R-D cost statistics that are estimated *on the fly*, allowing the proposed method to adapt to different video contents, encoder configurations and target complexity requirements, which can vary throughout the coding process.

Our proposal has been extensively tested, and the results prove that it works effectively for a wide variety of sequences and complexity requirements, outperforming the results achieved by a state-of-the-art complexity control method in terms of both accuracy in reaching the target complexity and coding efficiency, and also illustrating the advantages of the complexity control approach compared with the more common complexity reduction approach. Moreover, we have shown that our proposal is able to adapt its behavior when the complexity requirements vary over time.

Chapter 6

Conclusions and Future Lines of Research

6.1 Conclusions

In this Thesis two complexity control methods have been proposed in the context of the latest video coding standards, H.264/AVC and HEVC. These standards achieve a high compression efficiency at the expense of a great computational burden due to the need to select an appropriate coding option from a very large set of candidates.

The algorithms presented in this Thesis aim to control the complexity, i.e., to adapt the video coding system behavior to the available computational resources, trying to maximize the compression performance given such resources. To properly manage this matter is critical to take into account the time-varying statistical properties of the sequences to be coded. To this purpose, the first step to reach this goal was to conduct a thorough statistical analysis of the behavior of the coding systems and, then, relying on this statistical basis, to develop the algorithms. Our proposals are focused on the partition size decision in the H.264/AVC standard and on the coding unit depth decision in the HEVC standard, as these stages are some of the most computationally demanding of these video coding standards.

The R-D costs have been found to be proper input features to feed the decision

systems in both standards. In general terms, the complexity control is achieved by updating the decision thresholds according to the available and consumed resources in the coding process. Furthermore, an *on the fly* estimation of the parameters that define the decision thresholds has been proposed, so that our proposals are able to follow the time-varying content inherent to video sequences and the time-varying resources available in every application.

Finally, the proposals were extensively tested to prove its efficiency, comparing them with some methods of the state-of-the-art.

Regarding the specific contributions of this Thesis to the complexity control in H.264/AVC, they can be summarized as follows:

1. The main features of the proposed method are the following:

    - A hypothesis testing has been used to decide between low- and high-complexity models in a block basis.
    - R-D costs have been selected as input features to the hypothesis test.
    - Gaussian distributions have been assumed to make the problem tractable.
    - An *on the fly* estimation of the statistical parameters has been used to update the distributions.
    - An *on the fly* adaptation of the cost ratio of the hypothesis test has been proposed, according to the actual encoding times, to meet the complexity target.

2. As a result, the proposed algorithm fulfills the following requirements:

    - Negligible bit rate increments for high and medium complexities and acceptable bit rate increments for very low complexities.
    - Capability to reach any target complexity with higher accuracy and a better trade-off between complexity reduction and coding efficiency compared with other state-of-the-art approach.
    - Capability to adapt to a time-varying complexity target and video content.

- Capability to operate on a large dynamic range of target complexities with low miss-adjustment error.
- Proper performance with any spatial resolution.
- Low computational burden of the algorithm.

Concerning the specific contributions to the complexity control in HEVC:

1. This method is based on the following premises and features:
   - A set of early termination conditions have been designed at each CU depth level.
   - The early termination conditions have been designed as a set of thresholds, which are adjusted dynamically.
   - R-D costs are selected as input features to the statistical decision makers.
   - An *on the fly* estimation of the statistical parameters has been used to adjust the thresholds.
   - An *on the fly* adaptation of the thresholds has been proposed, according to the actual encoding times, to meet the complexity target.
2. The following requirements have been met by virtue of the previous features:
   - Significant complexity savings have been obtained in exchange for very limited losses in coding performance.
   - The results achieved by our method outperform those of a state-of-the-art proposal in terms of both accuracy in reaching the target complexity and coding efficiency.
   - Excellent performance for a wide variety of sequences and complexity requirements.
   - Capability to adapt its behavior when the complexity requirements or the video content vary over time.
   - High accuracy meeting a wide range of target complexities.

- Low computational burden of the algorithm.

Finally, it should be highlighted that the proposed methods turn out to be very adequate for a wide variety of applications, such as video-conference, where the performance of the video coding system must change with the time-dependent network conditions, or a mobile video application, where the resources of the mobile device changes depending on the number of applications running at the same time.

6.2 Future Lines of Research

We have identified several interesting future lines of research. First, the use of more complex classifiers to achieve higher accuracy in the fast decision processes is one of the most promising possibilities. Though we have already done some work in this direction, e.g. [Martinez-Enriquez et al., 2011] (in the context of complexity reduction), there is still room for improvement. Neural networks, support vector machines, or random forests are only a few examples of classifiers that could be implemented in this framework.

Other engaging future line of research would be to extend the complexity management design to take early decisions at other coding stages. In particular, in both standards the motion estimation process could also be addressed in a future design. Moreover, in the system designed for the HEVC standard, it would be interesting to extend our design to act at the PU and TU decision levels, which would undoubtedly lead to additional computational savings.

Furthermore, it would be possible to extend these proposals to work with other coding patterns including B-frames.

Finally, we could also look for alternative manners to mitigate the quality losses derived from limiting the coding options. To achieve this goal, filters could be developed, similar to the deblocking or SAO filters included in the standards, or to that proposed in [Jiménez-Moreno et al., 2014], since they are focused on reducing the artifacts caused by the coding process.

Bibliography

[Ahn et al., 2015] Ahn, S., Lee, B., and Kim, M. (2015). A Novel Fast CU Encoding Scheme Based on Spatiotemporal Encoding Parameters for HEVC Inter Coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(3):422–435.

[Ates and Altunbasak, 2008] Ates, H. and Altunbasak, Y. (2008). Rate-Distortion and Complexity Optimized Motion Estimation for H.264 Video Coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(2):159 –171.

[Blasi et al., 2013] Blasi, S., Peixoto, E., and Izquierdo, E. (2013). Enhanced inter-prediction using Merge Prediction Transformation in the HEVC codec. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 1709–1713.

[Bossen, 2011] Bossen, F. (2011). JCTVC-F900 Common test conditions and software reference configurations. Fifth meeting of the Joint Collaborative Team on Video Coding (JCT-VC) Torino, IT, 2011.

[Choi and Jang, 2012] Choi, K. and Jang, E. (2012). Early TU decision method for fast video encoding in high efficiency video coding. *Electronics Letters*, 48(12):689–691.

[Choi et al., 2011] Choi, K., Park, H.-M., and Jang, E. S. (2011). JCTVC-F092 Coding tree pruning based CU early termination. Fifth meeting of the Joint Collaborative Team on Video Coding (JCT-VC) Torino, IT, 2011.

[Choi et al., 2003] Choi, W. I., Jeon, B., and Jeong, J. (2003). Fast motion estimation with modified diamond search for variable motion block sizes. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 2, pages II – 371–4 vol.3.

[Correa et al., 2013] Correa, G., Assuncao, P., Agostini, L., and Cruz, L. (2013). Coding Tree Depth Estimation for Complexity Reduction of HEVC. In *Data Compression Conference (DCC), 2013*, pages 43–52.

[Correa et al., 2011] Correa, G., Assuncao, P., Agostini, L., and da Silva Cruz, L. (2011). Complexity control of high efficiency video encoders for power-constrained devices. *Consumer Electronics, IEEE Transactions on*, 57(4):1866–1874.

[da Fonseca and de Queiroz, 2009] da Fonseca, T. and de Queiroz, R. (2009). Complexity-constrained H.264 HD video coding through mode ranking. In *Picture Coding Symposium, 2009. PCS 2009*, pages 1 –4.

[da Fonseca and de Queiroz, 2011] da Fonseca, T. A. and de Queiroz, R. L. (2011). Complexity-constrained rate-distortion optimization for H.264/AVC video coding. In *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, pages 2909 –2912.

[de Suso et al., 2014] de Suso, J. L. G., Jiménez-Moreno, A., Martínez-Enríquez, E., and de María, F. D. (2014). Improved Method to Select the Lagrange Multiplier for Rate-Distortion Based Motion Estimation in Video Coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(3):452–464.

[Everett, 1963] Everett, H. (1963). Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources. *Operations Research*, 11(3):399–417.

[Gao et al., 2010] Gao, X., Lam, K. M., Zhuo, L., and Shen, L. (2010). Complexity scalable control for H.264 motion estimation and mode decision under energy constraints. *Signal Processing*, 90(8):2468 – 2479.

[G.Bjontegaard, 2001] G.Bjontegaard (2001). Calculation of Average PSNR Differences Between RD-Curves. ITU-T, VCEG-M33, Apr. 2001.

[Gonzalez-Diaz and Diaz-de Maria, 2008] Gonzalez-Diaz, I. and Diaz-de Maria, F. (2008). Adaptive Multipattern Fast Block-Matching Algorithm Based on Motion Classification Techniques. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(10):1369 –1382.

[Grecos and Mingyuan, 2006] Grecos, C. and Mingyuan, Y. (2006). Fast Mode Prediction for the Baseline and Main Profiles in the H.264 Video Coding Standard. *Multimedia, IEEE Transactions on*, 8(6):1125 –1134.

[Grellert et al., 2013] Grellert, M., Shafique, M., Karim Khan, M., Agostini, L., Mattos, J., and Henkel, J. (2013). An adaptive workload management scheme for HEVC encoding. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 1850–1854.

[G.Sullivan, 2001] G.Sullivan (2001). Recommended Simulation Common Conditions for H.26L coding efficiency Experiments on Low-Resolution Progressive-scan source material. ITU-T, VCEG-N81, Sept. 2001.

[Gweon et al., 2011] Gweon, R. H., Lee, Y.-L., and Lim, J. (2011). JCTVC-F045 Early termination of CU encoding to reduce HEVC complexity. Fifth meeting of the Joint Collaborative Team on Video Coding (JCT-VC) Torino, IT, 2011.

[Huijbers et al., 2011] Huijbers, E., Ozcelebi, T., and Bril, R. (2011). Complexity scalable motion estimation control for H.264/AVC. In *Consumer Electronics (ICCE), 2011 IEEE International Conference on*, pages 49 –50.

[ISO, 1993] ISO (1993). Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s. - Part 2: Video. *ISO/IEC 11172-2 MPEG-1*.

[ISO, 1998] ISO (1998). Information technology - Coding of audio-visual objects - Part 2: Visual. *ISO/IEC 14496-2*.

[ISO, 2013] ISO (2013). Information technology - High efficiency coding and media delivery in heterogeneous environments - Part 2: High efficiency video coding. *ISO/IEC 23008-2*.

[ISO/IEC, 1995] ISO/IEC (1995). Generic coding of Moving pictures and associated audio information - Part 2: Video. *ITU-T Recommendation H.262-ISO/IEC 13818-2, MPEG-2*.

[ITU-T, 1993] ITU-T (1993). ITU-T Recommendation H.261: Line Transmission of Non-Telephone Signals: Video Codec for Audiovisual Services at p x 64 kbts. *International Telecommunications Union*.

[ITU-T, 1998] ITU-T (1998). ITU-T Recommendation H.263: Video coding for low bit rate communications. *International Telecommunications Union*.

[ITU-T, 2003] ITU-T (2003). ITU-T Recommendation H.264: Advanced video coding for generic audiovisual services. *International Telecommunications Union*.

[Jimenez-Moreno et al., 2013] Jimenez-Moreno, A., Martinez-Enriquez, E., and Diaz-de Maria, F. (2013). Mode Decision-Based Algorithm for Complexity Control in H.264/AVC. *Multimedia, IEEE Transactions on*, 15(5):1094–1109.

[Jiménez-Moreno et al., 2016] Jiménez-Moreno, A., Martínez-Enríquez, E., and de María, F. D. (2016). Complexity Control Based on a Fast Coding Unit Decision Method in the HEVC Video Coding Standard. *IEEE Transactions on Multimedia*, 18(4):563–575.

[Jiménez-Moreno et al., 2014] Jiménez-Moreno, A., Martínez-Enríquez, E., Kumar, V., and de María, F. D. (2014). Standard-Compliant Low-Pass Temporal Filter to Reduce the Perceived Flicker Artifact. *IEEE Transactions on Multimedia*, 16(7):1863–1873.

[Kannangara et al., 2009] Kannangara, C., Richardson, I., Bystrom, M., and Zhao, Y. (2009). Complexity Control of H.264/AVC Based on Mode-Conditional Cost Probability Distributions. *Multimedia, IEEE Transactions on*, 11(3):433 –442.

[Kannangara et al., 2008] Kannangara, C., Richardson, I., and Miller, A. (2008). Computational Complexity Management of a Real-Time H.264/AVC Encoder. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(9):1191 –1200.

[Kim et al., 2014] Kim, J., Lee, J. H., Kim, B. G., and Choi, J. S. (2014). Fast mode decision scheme using sum of the absolute difference-based Bayesian model for the H.264/AVC video standard. *IET Signal Processing*, 8(5):530–539.

[Kim et al., 2012] Kim, J., Yang, J., Won, K., and Jeon, B. (2012). Early determination of mode decision for HEVC. In *Picture Coding Symposium (PCS), 2012*, pages 449–452.

[Kraskov et al., 2004] Kraskov, A., Stögbauer, H., and Grassberger, P. (2004). Estimating mutual information. *Phys. Rev. E*, 69(6):066138.

[Kuo and Chan, 2006] Kuo, T.-Y. and Chan, C.-H. (2006). Fast Variable Block Size Motion Estimation for H.264 Using Likelihood and Correlation of Motion Field. *Circuits and Systems for Video Technology, IEEE Transactions on*, 16(10):1185 –1195.

[Lee et al., 2015] Lee, J., Kim, S., Lim, K., and Lee, S. (2015). A Fast CU Size Decision Algorithm for HEVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(3):411–421.

[Leng et al., 2011] Leng, J., Sun, L., Ikenaga, T., and Sakaida, S. (2011). Content Based Hierarchical Fast Coding Unit Decision Algorithm for HEVC. In *Multimedia and Signal Processing (CMSP), 2011 International Conference on*, volume 1, pages 56–59.

[Li et al., 2014] Li, C., Wu, D., and Xiong, H. (2014). Delay-Power-Rate-Distortion Model for Wireless Video Communication Under Delay and Energy Constraints. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(7):1170–1183.

[Li et al., 2005] Li, G.-L., Chen, M.-J., Li, H.-J., and Hsu, C.-T. (2005). Efficient search and mode prediction algorithms for motion estimation in H.264/AVC. In *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pages 5481 – 5484 Vol. 6.

[Li et al., 2009] Li, X., Oertel, N., Hutter, A., and Kaup, A. (2009). Laplace Distribution Based Lagrangian Rate Distortion Optimization for Hybrid Video Coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(2):193–205.

[Lu and Martin, 2013] Lu, X. and Martin, G. R. (2013). Fast Mode Decision Algorithm for the H.264/AVC Scalable Video Coding Extension. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(5):846–855.

[Martinez-Enriquez et al., 2007] Martinez-Enriquez, E., de Frutos-Lopez, M., Pujol-Alcolado, J., and Diaz-de Maria, F. (2007). A Fast Motion-Cost Based Algorithm for H.264/AVC Inter Mode Decision. *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, 5:V –325–V –328.

[Martinez-Enriquez et al., 2011] Martinez-Enriquez, E., Jimenez-Moreno, A., Angel-Pellon, M., and Diaz-de Maria, F. (2011). A Two-Level Classification-Based Approach to Inter Mode Decision in H.264/AVC. *Circuits and Systems for Video Technology, IEEE Transactions on*, 21(11):1719 –1732.

[Martinez-Enriquez et al., 2009] Martinez-Enriquez, E., Jimenez-Moreno, A., and Diaz-de Maria, F. (2009). A novel fast inter mode decision in H.264/AVC based on a regionalized hypothesis testing. In *Picture Coding Symposium, 2009. PCS 2009*, pages 217–220.

[Martinez-Enriquez et al., 2010] Martinez-Enriquez, E., Jimenez-Moreno, A., and Diaz-de Maria, F. (2010). An adaptive algorithm for fast inter mode decision in the H.264/AVC video coding standard. *Consumer Electronics, IEEE Transactions on*, 56(2):826 –834.

[Richardson, 2003] Richardson, I. (2003). H.264 and MPEG-4 video compression. *Wiley*.

[Saha et al., 2007] Saha, A., Mallic, K., Mukherjee, J., and Sural, S. (2007). SKIP Prediction for Fast Rate Distortion Optimization in H.264. *Consumer Electronics, IEEE Transactions on*, 53(3):1153 –1160.

[Shen et al., 2013] Shen, L., Liu, Z., Zhang, X., Zhao, W., and Zhang, Z. (2013). An Effective CU Size Decision Method for HEVC Encoders. *Multimedia, IEEE Transactions on*, 15(2):465–470.

[Shen et al., 2012] Shen, X., Yu, L., and Chen, J. (2012). Fast coding unit size selection for HEVC based on Bayesian decision rule. In *Picture Coding Symposium (PCS), 2012*, pages 453–456.

[Su et al., 2009] Su, L., Lu, Y., Wu, F., Li, S., and Gao, W. (2009). Complexity-Constrained H.264 Video Encoding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(4):477 –490.

[Sullivan and Wiegand, 1998] Sullivan, G. and Wiegand, T. (1998). Rate-distortion optimization for video compression. *Signal Processing Magazine, IEEE*, 15(6):74 –90.

[Tan et al., 2012] Tan, H. L., Liu, F., Tan, Y. H., and Yeo, C. (2012). On fast coding tree block and mode decision for High-Efficiency Video Coding (HEVC). In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 825–828.

[Tan et al., 2010] Tan, Y. H., Lee, W. S., Tham, J. Y., Rahardja, S., and Lye, K. M. (2010). Complexity Scalable H.264/AVC Encoding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(9):1271 –1275.

[Teng et al., 2011] Teng, S.-W., Hang, H.-M., and Chen, Y.-F. (2011). Fast mode decision algorithm for Residual Quadtree coding in HEVC. In *Visual Communications and Image Processing (VCIP), 2011 IEEE*, pages 1–4.

[Tourapis and Tourapis, 2003] Tourapis, H.-Y. and Tourapis, A. (2003). Fast motion estimation within the H.264 codec. In *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, volume 3, pages III – 517–20 vol.3.

[Ukhanova et al., 2013] Ukhanova, A., Milani, S., and Forchhammer, S. (2013). Game-theoretic rate-distortion-complexity optimization for HEVC. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 1995–1999.

[Vanam et al., 2007] Vanam, R., Riskin, E., Hemami, S., and Ladner, R. (2007). Distortion-Complexity Optimization of the H.264/MPEG-4 AVC Encoder using the GBFOS Algorithm. In *Data Compression Conference, 2007. DCC '07*, pages 303–312.

[Vanam et al., 2009] Vanam, R., Riskin, E., and Ladner, R. (2009). H.264/MPEG-4 AVC Encoder Parameter Selection Algorithms for Complexity Distortion Tradeoff. In *Data Compression Conference, 2009. DCC '09.*, pages 372 –381.

[Vanne et al., 2012] Vanne, J., Viitanen, M., Hamalainen, T., and Hallapuro, A. (2012). Comparative Rate-Distortion-Complexity Analysis of HEVC and AVC Video Codecs. *Circuits and Systems for Video Technology, IEEE Transactions on*, 22(12):1885–1898.

[Xiong et al., 2014] Xiong, J., Li, H., Wu, Q., and Meng, F. (2014). A Fast HEVC Inter CU Selection Method Based on Pyramid Motion Divergence. *Multimedia, IEEE Transactions on*, 16(2):559–564.

[You et al., 2006] You, J., Kim, W., and Jeong, J. (2006). 16x16 macroblock partition size prediction for H.264 P slices. *Consumer Electronics, IEEE Transactions on*, 52(4):1377 –1383.

[Yusuf et al., 2014] Yusuf, M. S. U., Hossain, M. I., and Ahmad, M. (2014). Probability based fast inter mode decision algorithm for H.264/AVC video standard. In *Strategic Technology (IFOST), 2014 9th International Forum on*, pages 52–55.

[Zhang et al., 2003] Zhang, J., He, Y., Yang, S., and Zhong, Y. (2003). Performance and complexity joint optimization for H.264 video coding. In *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on*, volume 2, pages II–888 – II–891 vol.2.

[Zhang et al., 2013] Zhang, Y., Wang, H., and Li, Z. (2013). Fast Coding Unit Depth Decision Algorithm for Interframe Coding in HEVC. In *Data Compression Conference (DCC), 2013*, pages 53–62.

[Zhao et al., 2013] Zhao, T., Wang, Z., and Kwong, S. (2013). Flexible Mode Selection and Complexity Allocation in High Efficiency Video Coding. *Selected Topics in Signal Processing, IEEE Journal of*, 7(6):1135–1144.

[Zhou et al., 2009] Zhou, C., Tan, Y., Tian, J., and Lu, Y. (2009). 3 σ-rule-based early termination algorithm for mode decision in H.264. *Electronics Letters*, 45(19):974 –975.

[Zhu et al., 2002] Zhu, C., Lin, X., and Chau, L.-P. (2002). Hexagon-based search pattern for fast block motion estimation. *Circuits and Systems for Video Technology, IEEE Transactions on*, 12(5):349 –355.