**Ecole Polytechnique de Louvain**
Laboratoire de Télécommunications
et
Télédétection

B - 1348 Louvain-la-Neuve
Belgique

# JPEG 2000 and Parity Bit Replenishment for Remote Video Browsing

François-Olivier Devaux

*Thèse présentée en vue de l'obtention du grade de
docteur en sciences de l'Ingénieur*

Composition du jury :

Christophe De Vleeschouwer (UCL/TELE) - *Promoteur*
Pascal Frossard (EPFL, Suisse)
Gauthier Lafruit (IMEC, Belgique)
Benoit Macq (UCL/TELE)
Luc Vandendorpe (UCL/TELE) - *Président*

Septembre 2008

# Acknowledgments

I would like to express my sincere gratitude to my supervisor Christophe De Vleeschouwer for the guidance, advice and encouragements he provided all along this PhD. I am so thankful for his incredible support, availability, enthusiasm and dedication. By the way, Christophe, I am very proud to be the first PhD student you have supervised !

I have been lucky enough to receive the support of another supervisor, Benoit Macq, during the first year of my PhD. Thank you Benoit for creating this opportunity to study exciting and fascinating research topics in the image processing field. Thank you for the impressive energy you radiate in the laboratory and for the research contacts and collaborations you have made possible. Thank you also for supporting the intoPIX project since the first day.

I would like to thank the other members of my jury, Pascal Frossard, Gauthier Lafruit and Luc Vandendorpe for their high quality advice and comments.

It was a pleasure during the last four years to share my office with many distinguished colleagues. Thank you Antonin, Damien, Jean-François, Cédric, Joanna, Rémy, Parvatha, Max and Andrew. I also want to mention the valuable friendship that was build with many other researchers during these years, with a special mention to Pedro, Jacek, Jean-Julien, Jerome, . . . I am thankful to all present and past members of the Communication and Remote Sensing Laboratory for making it a stimulating and pleasant environment.

I am very grateful to the intoPIX founders and team. It has been a great pleasure to participate with you in this great adventure that has just started.

I would like to thank my parents and family for their everlasting love and care. Thank you mom and dad for your incredible support during all these years.

Last but most importantly, thank you Pascaline and Gilles, my two sunshines. You fill my life with so much love and affection… Thank you Pascaline for your patience and support during these four years, and thank you Gilles for helping me out with the $c0_{mp}lic^{a}Ted$ equations.

# Abstract

THis thesis is devoted to the study of a compression and transmission framework for video. It exploits the JPEG 2000 standard and the coding with side information principles to enable an efficient interactive browsing of video sequences.

During the last decade, we have witnessed an explosion of digital visual information as well as a significant diversification of visualization devices. In terms of viewing experience, many applications now enable users to interact with the content stored on a distant server. Pausing video sequences to observe details by zooming and panning or, at the opposite, browsing low resolutions of high quality HD videos are becoming common tasks. The video distribution framework envisioned in this thesis targets such devices and applications.

Based on the conditional replenishment framework, the proposed system combines two complementary coding methods. The first one is JPEG 2000, a scalable and very efficient compression algorithm. The second method is based on the coding with side information paradigm. This technique is relatively novel in a video context, and has been adapted to the particular scalable image representation adopted in this work. Interestingly, it has been improved by integrating an image source model and by exploiting the temporal correlation inherent to the sequence.

A particularity of this work is the emphasis on the system scalability as well as on the server complexity. The proposed browsing architecture can scale to handle large volumes of content and serve a possibly very large number of heterogeneous users. This is achieved by defining a scheduler that adapts its decisions to the channel conditions and to user requirements expressed in terms of computational capabilities and spatio-temporal interest. This scheduling is carried out in real-time at low computational cost and in a post-compression way, without re-encoding the sequences.

# Contents

# List of Figures

# Introduction 1

## 1.1 Motivation

During the last decade, we have witnessed an explosion of digital visual information. The number of devices acquiring multimedia data continues to increase exponentially, with an ever finer quality [24]. With the price of storage virtually tending to zero and an increasing part of the worldwide population gaining access to the Internet, this content can potentially reach an incredible number of users at anytime and almost anywhere.

At the same time, the variety of visualization devices is very wide: the same content can be played on PDAs with small tactile screens up to very large displays of personal desktops. These devices also differ in their streaming capability and computation resources. For example, nowadays, a new movie is typically first played in digital cinemas with 2K displays[1] streamed through Gigabit networks and decoded on powerful cinema servers, and ends up a few month later on small Ipod screens, streamed through the Internet and decoded with limited chips.

In terms of viewing experience, many applications now enable users to interact with the content. Pausing the video sequence to observe details by zooming and cropping or, at the opposite, browsing low resolutions of high quality HD videos are becoming common tasks. Such interactive actions are particularly essential in a video-surveillance context on which we have focused in this work. An efficient browsing of these video-surveillance scenes increases the quantity and quality of information extracted.

This thesis is devoted to the study of compression and transmission

[1] A 2K resolution corresponds to 2048x1080 pixels. The most recent digital cinemas are equipped with 4K screens (4096x2160 pixels).

techniques enabling an efficient browsing of video content. A particularity of this work is the emphasis on the system scalability as well as on the server limited complexity. In addition, the proposed framework enables end-user resources and interest with regard to the displayed scene to be taken into account in real-time by adapting the streamed content, without re-encoding the sequences.

## 1.2 Problem statement and contributions

We consider application scenarios in which a client - typically a human controller behind a PC or a wireless PDA - accesses pre-encoded content captured by possibly multiple (overlapping) surveillance cameras, to figure out what happened in the monitored scene at some earlier time. A desired browsing interface enables the end-user to randomly select any spatio-temporal segment of the video(s) at arbitrary resolution so that, in a typical interactive browsing scenario, the end-user can first survey the (multiple) video(s) at low temporal and spatial resolution, and thereafter focus on higher resolution displays of short video segments of interest, or decide to zoom in on a specific spatial area or object of interest, either in a particular frame or video segment.

Regarding deployment, we are interested in a browsing architecture that can scale to handle large volumes of content, captured by several cameras, on multiple sites and at distinct time instants. Therefore, the content has to be stored efficiently in a compressed format and the computational load associated to content storage, access and distribution has to be limited[1].

To address the above requirements of scalability and deployment at large scale, we have decided to build our system on the image representation defined in the JPEG 2000 compression standard [3]. JPEG 2000 indeed provides a natural solution to support the required access flexibility, through low complexity manipulation of pre-encoded bitstreams [41][14], without requiring computationally expensive transcoding -i.e. decompression followed by compression- operations. In the meantime, we have also renounced to exploit temporal prediction during compression, in order to preserve the capability of random temporal access to each individual frame of the sequence.

[1] Even in cases for which a given content is only accessed by a few clients, the server is expected to handle a large number of contents simultaneously, thereby making computational load on the server an important issue.

To mitigate the penalty induced by a strict INTRA coding structure, we have introduced an innovative replenishment method based on JPEG 2000 and parity bit coding, and have adapted conditional replenishment principles to scalable representations and to the specificities of video surveillance scenes. This adaptation has been achieved at two levels. First, next to the previously reconstructed frame, the pre-computed estimation of the scene background has been considered as a potential candidate for reconstructing the current frame in absence of replenishment information. Secondly, decisions about the replenishment of JPEG 2000 and parity packets have been optimized in the RD sense by taking into account potential semantic information, e.g. defining some knowledge about the regions interesting the user in the scene. Interestingly, that knowledge is exploited independently of the compression stage, which means that it can be provided a posteriori, at transmission time by each individual user. Hence, our system naturally supports interactive definition of windows of interest to specify the fraction of a pre-encoded content that (s)he wants to visualize at a given moment during the streaming.

Finally, the integrated contributions of our work result in a video server which:

- implements a multi-reference conditional replenishment scheme for pre-encoded JPEG 2000, and demonstrates the relevance of the approach in scenarios for which the video sequence is captured with still cameras, as often encountered in a video surveillance context;
- makes use of an image source model and exploits the temporal correlation in the sequence to improve the correcting capabilities of the parity-based refresh mechanism;
- promotes adaptive and user-driven access to video content by defining a scheduler adapting to heterogeneous channel conditions and to user requirements (in terms of spatio-temporal interest) at low computational cost and in a post-compression way, based on a set of pre-calculated rate distortion metadata;
- circumvents the drawbacks of closed-loop prediction systems by restricting transmissions to INTRA and parity bit content. This is especially relevant when addressing heterogeneous clients dealing with different prediction references in lossy environments;

- does not aim at competing with state-of-the-art hybrid video compression algorithms [51] [66]. Instead of compression efficiency, our proposed solution emphasizes the capabilities for adaptation to user preference, and spatio-temporal random access required for interactive navigation through the individual frames or segments of the video sequence.

## 1.3 Outline of the thesis

Besides this introductory chapter, our thesis is divided into five chapters as follows.

In Chapter 2, we present the state of the art in video coding and justify the options we have chosen to develop our video sever. In particular, we review the INTRA-based conditional replenishment mechanisms and parity-based video coding techniques. We also propose an overview of the scalable coding systems and focus on the image compression standard JPEG 2000. The remainder of this thesis consists in presenting how parts of these different techniques have been combined in a coherent framework to enable an efficient remote browsing of video content.

In Chapter 3, the proposed video codec supporting our flexible and interactive server is presented. We detail the adaptation of the conditional replenishment principle to the scalable representations derived from JPEG 2000 and parity coding, and explain how an optimal allocation of the compressed data in a rate-distortion sense can be achieved. At transmission time, individual user requirements and interests in parts of the content, like regions of interest, influence the way this rate-allocation process schedules the pre-encoded content. Simulations demonstrate the efficiency and flexibility of the system.

Chapter 4 further details the generation and decoding of parity bits based on the theory of coding with side information. Spatial and temporal correlation are exploited in the wavelet domain and lead to a significant improvement of the system performances. Important lessons are drawn about the exploitation of video source models in parity-based coding paradigms. In particular, it is shown that a localized prediction error can be more easily corrected than an error with similar energy that is spread over the whole image. This learning should certainly drive the design of an ad-hoc motion compensation engine.

In Chapter 5, we integrate the conditional replenishment framework into a server supplying a large number of heterogeneous users with different resources and requirements. We show that simple approximations can greatly reduce the server complexity when adapting the pre-encoded content to each user needs and resources.

In the final chapter, we summarize the contributions of our work and suggest several areas of interest for future research.

# 2 State of the art in video coding

*An efficient exploitation of the temporal correlation intrinsic to video sequences is a requirement in order to reduce bandwidth consumption when streaming video content. Most video coding algorithms rely on closed-loop prediction to achieve high compression efficiency. This efficiency is counterbalanced by strong dependencies within the compressed content. These constraints are unacceptable for some specific applications requiring a higher flexibility in the way they access the compressed content. For this reason, a large effort has been made in the last decade to alleviate the closed-loop constraint by multiplying the prediction paths within a single scalable and embedded bitstream, each path corresponding to a distinct decoding of the codestream. In this section, we review these advances in scalable video coding. As an introduction to Chapter 3 and 4 of this thesis, we also present alternative coding and transmission mechanisms to exploit temporal redundancy while avoiding closed-loop prediction.*

## 2.1 Introduction

In this chapter, we present different video coding systems and compare the way they exploit temporal correlation. In particular, we are interested in the consequences of closed-loop predictions. The basic principle of coding mechanisms integrating prediction loops consists in dividing the video sequence into groups of frames, and encoding independently the first frame, called INTRA frame. For each one of the following frames of the group, called INTER frames, the prediction error of a motion compensated version of the previous frame is recursively encoded, as illustrated in Figure 2.1. The dependency between INTRA and INTER frames is depicted in Figure 2.2. Todays most efficient video coding systems integrate prediction loops which are improved versions of this basic principle (e.g. predictions based on several previous or subsequent references, scalable multiple-layer structure, ...).

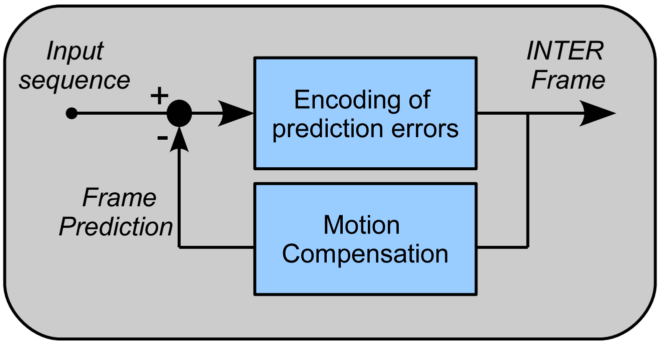


Figure 2.1: *Basic principle of closed-loop coding. The prediction error of a motion compensated version of the previous frame is recursively encoded to generate INTER frames. The predicted frame is also called reference.*

While prediction loops exploit temporal correlation very efficiently, they present several drawbacks. First, a perfect synchronization is required between the encoder and the decoder to ensure that they both consider the same references. In particular, any error in the reference propagates to all subsequent INTER frames. Second, this tight synchronization prevents a real-time adaptation of the content to the client resources and needs. Finally, prediction loops prevent an efficient random access to frames, as the INTRA reference and all intermediary frames from which the targeted frame is predicted must be decoded first. Hence, when an access to individual frames is expected, closed-loop systems are characterized by an intrinsic latency and a low coding efficiency since several frames must be transmitted and decoded to output a single frame.

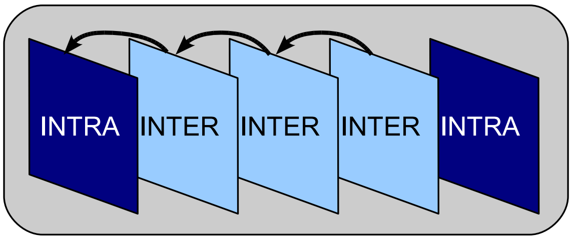


Figure 2.2: *Frame dependencies with a basic prediction structure.*

The penalty induced by these drawbacks depends on the application at hand. In this work, we focus on applications that require a random access to individual frames, and for which a pre-encoded stream is likely to serve heterogeneous clients, manipulating distinct references. In this context, we will evaluate alternative mechanisms to closed-loop predictions to exploit temporal correlation.

This chapter is structured as follows. We first define scalable coding in an image and video context. We then present the JPEG 2000 still image coding standard which offers a flexible and scalable image structure and is at the root of our proposed video coding framework. Next, we propose an overview of SVC, the state of the art video coding standard which offers a high coding efficiency, combined with a rich scalability. Then, we present the conditional replenishment principles, on which our proposed system is based. Video coding systems based on conditional replenishment are the first systems which are not based on closed-loop prediction. We finally present video coding with side information, a novel video coding paradigm, and explain how it relaxes the closed loop constraint.

## 2.2 Scalable image and video coding

*Scalable coding* refers to the ability to create a single embedded bitstream from which several versions of the content can be extracted. In an image coding context, a scalable codec enables users to extract from the compressed image lower resolutions, limited spatial zones, lower quality versions, or a reduced number of components. This is achieved by decoding only to the parts of interest within the compressed data. In a video context, additional levels of scalability consist in extracting lower temporal resolutions of the content, and in accessing randomly selected frames from the video stream without decoding adjacent frames.

The main motivation to offer scalable coding systems is to enable a unique compressed content to be distributed to clients with different resources and requirements. The content can be adapted to clients processing power, displays, network capabilities as well as semantic interest in the content. This adaptation can be achieved in real-time by a server transmitting only parts of the video bitstream corresponding to the clients ressources.

Scalable video coding has been an active research and standardization area for at least 20 years. The international video coding standards H.262/MPEG-2 Video [31], H.263 [29], and MPEG-4 Visual [28] already included several scalability tools, which have however been rarely used mainly because of the loss in coding efficiency associated with these scalable modes and the increased decoder complexity. The recent scalable extension of H.264/AVC [30] mitigates these drawbacks and is presented in the next section.

It is worth mentioning that all these video standards follow the same general closed-loop predictive compression scheme that we now summarize. Specifically, each picture is partitioned into macroblocks which are either spatially or temporally predicted based on neighboring macroblocks. The remaining correlation of the resulting prediction residuals is exploited by applying a transform such as the DCT followed by quantization and entropy coding. We will see in Section 2.4 that the closed-loop paradigm severely constraints the construction and exploitation of scalable codestreams.

## 2.3 JPEG 2000

Although the context of our work is video, we now present the JPEG 2000 still image standard. Its flexible image representation structure will be the core of our system, and deserves a presentation in this chapter.

In the following, we first give an overview of the main coding steps of the JPEG 2000 algorithm and then present the various scalable facets of its image representation. Most parts of the algorithm presented here are integrated in our coding system. Finally, we give a brief overview of the fields in which the JPEG 2000 compression is deployed today.

### 2.3.1 Wavelet domain

According to the JPEG 2000 standard [3], a first Discrete Wavelet Transform (DWT) is applied on the original image, generating four *subbands* (LL, HL, LH and HH) containing the vertical and horizontal low (L) and high (H) frequencies of the original data. The DWT is then applied recursively on the LL subbands containing the low frequencies of each resolution, as illustrated in Figure 2.3. The subbands resulting from the wavelet transform are partitioned into *code-blocks* that are coded independently [3] [41] [16].

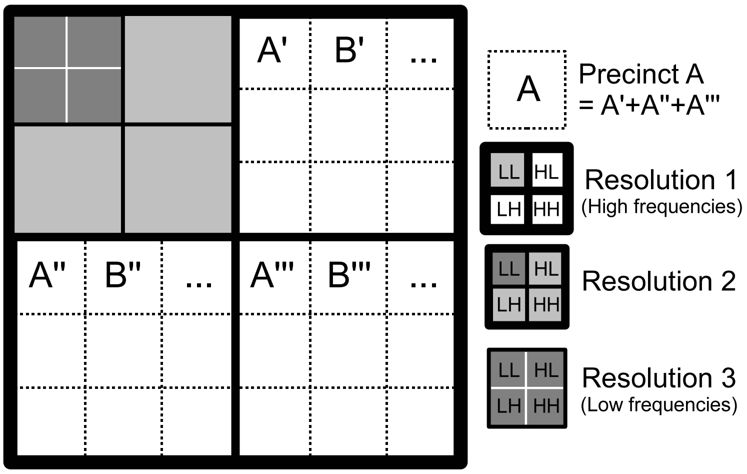


Figure 2.3: *JPEG 2000 wavelet subbands and precincts. A succession of discrete wavelet transforms are applied recursively to the original image. A precinct (e.g. Precinct A) is a spatial subdivision of a resolution, and corresponds to the same spatial zones in each subband (e.g.* $A' + A'' + A'''$*).*

### 2.3.2 Entropy coding

Each code-block is compressed independently using a *context based adaptive entropy coder*. It reduces the amount of data without losing information by removing redundancy from the original binary sequence. "Entropy" means it achieves this redundancy reduction by using the probability estimates of the symbols. Adaptability is provided by dynamically updating these probability estimates during the coding process. And "context-based" means that the probability estimate of a symbol depends on its neighborhood (its "context").

Practically, entropy coding consists of the following steps:

- *Context Modeling*: the code-block data are arranged in order to first encode the bits that contribute to the largest distortion reduction for the smallest increase in file size. In JPEG 2000, the Embedded Block Coding with Optimized Truncation (EBCOT) algorithm [67] has been adopted to implement this operation. It is based on the observation that spatially, the value of a bit inside a bit-plane can be estimated mainly on the value of its neighbors [52].

  The EBCOT algorithm scans code-blocks bit-plane after bit-plane in order to label the bits according to the value of their neighbors, characterized by their *context*. These 19 different contexts are computed based on state variables related to the 8 surrounding coefficients, and to the processed coefficient itself. The most important state variable is the *significance* status of a coefficient. A coefficient is said to switch from non-significant to significant state at the most significant bit-plane for which a bit equal to '1' is found for this coefficient. Other variables affecting the context are the type of subband (LL, HL, LH or HH), the sign of the coefficient, and its "first refinement" status[1].

- *Arithmetic Coding*: The outputs of the Context Modeling step are entropy coded using a MQ-coder, which is a derivative of the Q-coder [48]. According to the provided context, the coder chooses a probability for the bit to encode, among predetermined probability values supplied by the JPEG 2000 Standard and stored in a lookup table (LUT). Using this probability, it encodes the bit and progressively generates code-words, called segments, that will be organized to form the final code-stream.

Intuitively, it is easy to understand that a context improves the predictability of encoded binary values. Indeed, in a given bit-plane, if a non-significant coefficient is surrounded with significant ones, it is more likely to become significant (i.e. get a '1' bit) than if it was surrounded with non-significant coefficients. Similarly, if a coefficient has become significant in the previous bit-plane and is surrounded with non-significant coefficients only, there is a higher probability for the coefficient to lie in the lower half of the remaining potential values, and therefore for the very first refinement bit to be a '0'.

Figure 2.4 depicts the representation of a code-block in bit-planes and the concept of context.

### 2.3.3 Quality scalability

As explained before, code-blocks are described sequentially by bit-planes from most to least significant. This description by bit-planes plays a dual role in quantization and progressive transmission, by realizing a sequence

[1] This variable is always equal to '0', except at the bit-plane immediately following the bit-plane where the coefficient became significant, where it is set to '1'.

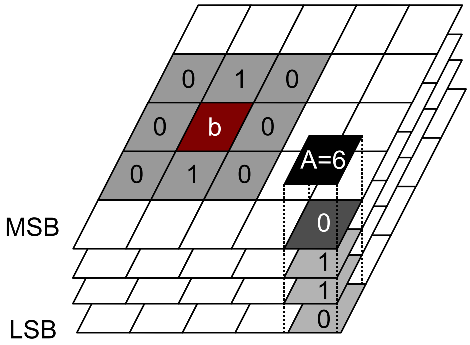


Figure 2.4: ***Bit-plane representation***. *The coefficient A which is equal to 6 is represented in binary form (0110) through the four bit-planes. Its MSB is non-significant and the remaining bits of the coefficient are significant.* ***Context***. *The context of bit b is calculated based on the significance value of its 8 neighbors and on its own significance state.*

of successively refined uniform quantizers [60]. The most significant bits from each coefficient provide a coarse idea of its final value and contribute therefore to a larger extent to the global distortion reduction than less significant bits.

Each code-block is coded into an embedded bitstream, i.e. into a stream that provides a representation that is (close-to-)optimal in the rate-distortion sense when truncated to any desired length. To achieve rate-distortion (RD) optimal scalability at the image level, the embedded bitstream of each code-block is partitioned into a sequence of increments based on a set of truncating points that correspond to the various rate-distortion trade-offs [67] defined by a set of Lagrange multipliers. A Lagrange multiplier $\lambda$ translates a cost in bytes in terms of distortion. It defines the relative importance of rate and distortion. Given $\lambda$, the RD optimal truncation of a code-block bitstream is obtained by truncating the embedded bitstream so as to minimize the Lagrangian cost function

$$\mathcal{L}(\lambda) = D(R) + \lambda R \tag{2.1}$$

where $D(R)$ denotes the distortion resulting from the truncation to $R$ bytes. Different Lagrange multipliers define different rate-distortion trade-offs, which in turn result in different truncation points.

For each code-block, a decreasing sequence of Lagrange multipliers $\{\lambda_q\}_{q>0}$ identifies an ordered set of truncation points that partition the code-block bitstream into a sequence of incremental contributions [67]. Incremental contributions from the set of image code-blocks are then collected into so-called quality layers, $\mathcal{Q}_q$. The targeted rate-distortion trade-offs during the truncation are the same for all the code-blocks. Consequently, for any quality layer index $l$, the contributions provided by layers $\mathcal{Q}_1$ through $\mathcal{Q}_l$ constitute a rate-distortion optimal representation of the entire image. It thus provides at the image level distortion scalability, also called quality scalability.

### 2.3.4 Spatial and resolution scalability

Although they are coded independently, code-blocks are not identified explicitly within a JPEG 2000 codestream. Instead, the code-blocks associated to a given resolution are grouped into *precincts*, based on their spatial location [3, 15]. Hence, a precinct corresponds to the parts of the JPEG 2000 codestream that are specific to a given resolution and spatial location. The concept of precinct is illustrated in Figure 2.3. Precincts sizes can vary for each resolution, but a certain coherence in the sizes throughout the resolutions is welcome to ensure a true spatial scalability. Typically, a spatial zone of size $(w, h)$ can be efficiently extracted from a codestream when the precincts have a size of $(w*2^{-r}, h*2^{-r})$ where $r$ correspond to the resolution number as illustrated in Figure 2.3 ($r = 3$ for the low resolution).

As JPEG 2000 packets are generated independently for each precinct, individual decisions can be taken for each precinct, offering the spatial and resolution scalability.

### 2.3.5 Codestream structure

As a consequence of the above-defined quality layering, a precinct can also be viewed as a hierarchy of *packets*, each packet collecting the parts of the codestream that correspond to a given quality among all code-blocks matching the precinct resolution and position. Hence, packets are the basic access unit in the JPEG 2000 codestream.

Besides packets and their associated packet header, JPEG 2000 codestreams contain main headers specifying the image characteristics and the JPEG 2000 parameters used, as depicted in Figure 2.5.

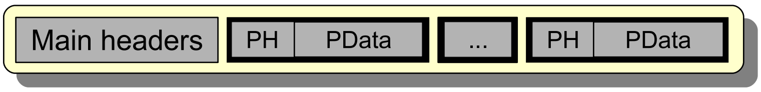


Figure 2.5: *The JPEG 2000 codestream contains main headers followed by a succession of packets composed of a Packet Header (PH) and Packet Data (PData).*

### 2.3.6 JPEG 2000 deployment

Nowadays, the JPEG 2000 standard is widely used in professional imaging fields which require high efficiency, high bit-depth and scalability [34]. The main applications using JPEG 2000 are digital cinema, video surveillance, geographic imaging, archiving and medical imaging.

The success of JPEG 2000 for video applications can be explained by the fact that at high bitrates[1], its compression efficiency is similar to that of H.264/AVC. Moreover, its high temporal and spatial scalability as well as its robustness to transmission errors are other features that make JPEG 2000 a good candidate for high quality video transmissions.

Although JPEG 2000 has been successfully deployed in professional applications, its adoption in consumer applications is far from being convincing. Its high complexity seems to be the main reason preventing such adoption.

## 2.4 SVC

SVC is the scalable extension of the H.264/AVC standard [30]. The objective of the SVC standardization [59] has been to enable the encoding of a high-quality video bitstream that contains one or more subset bit streams that can themselves be decoded with a complexity and reconstruction quality similar to that achieved using the H.264/AVC design with the same quantity of data as in the subset bitstream.

Regarding spatial scalability, SVC follows the conventional approach of multi-layer coding used in previous H.26x and MPEG standards. Figure 2.6 depicts a multi-layer structure with two layers. The lower layer, called *base*

[1] Depending on the content, it is considered that JPEG 2000 efficiency is comparable to that of H.264/AVC at bitrates above 100-150 Mbps for 2K (2048x1080 pixels) video sequences.

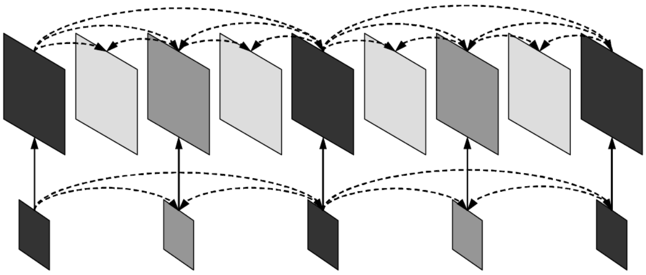


Figure 2.6: *Multi-layer structure adopted by the SVC standard, which enables spatial and quality scalability. The layer at the bottom of the figure corresponds to the base layer, which can be refined by the top enhancement layer. In this configuration, the enhancement layer increases the resolution and frame rate of the base layer. This figure has been published in the SVC overview paper by Wiegand et al. [59] and is reproduced here with the kind permission of the authors.*

*layer*, corresponds to the low spatial resolution and the second layer, called *enhancement layer*, corresponds to a higher resolution. Quality scalability can be considered as a special case of spatial scalability for which identical picture sizes for base and enhancement layers are considered. In this case, the base layer contains a quantized version of the coefficients which are refined by the enhancement layer.

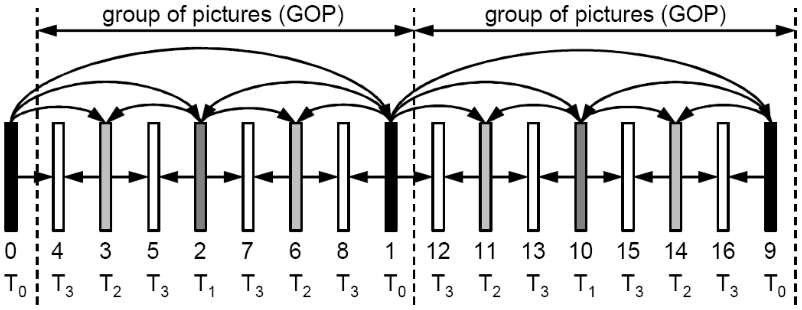


Figure 2.7: *SVC temporal scalability. Example of hierarchical prediction structure enabling temporal scalability. Four temporal resolutions ($T_0$...$T_3$) can be extracted with this structure. This figure has been published in the SVC overview paper by Wiegand et al. [59] and is reproduced here with the kind permission of the authors.*

SVC provides temporal scalability by partitioning the bitstream into a temporal base layer and one or more temporal enhancement layers. Figure 2.7 gives an example of SVC hierarchical prediction structure between temporal layers. At the bottom of the figure, the encoding order (0...16) is first specified followed by the layer number ($T_0$...$T_3$ where $T_0$ refers to the temporal base layer). A low frame-rate version of the sequence can be obtained by decoding only the $T_0$ frames. Spatial and temporal scalability can be combined by reducing the number of lower layer pictures, as depicted in Figure 2.6, where the base layer corresponds to a low frame-rate and resolution of the sequence.

SVC provides a high flexibility in its hierarchical prediction structure. For example, if applications require temporal scalability with low latency, the structure depicted in Figure 2.8 in which the prediction is only based on previous frames can be adopted instead of the structure of Figure 2.7. Such low latency structure is penalized by a lower coding efficiency due to a reduced prediction horizon.

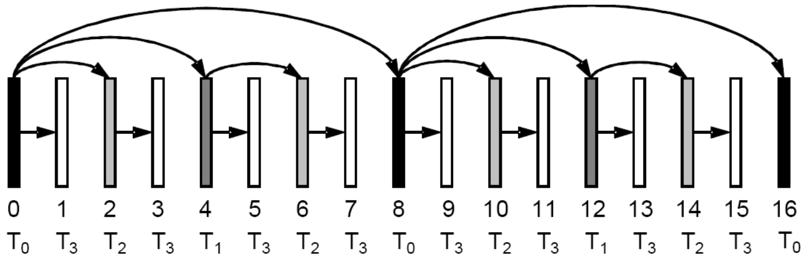


Figure 2.8: *Example of hierarchical prediction structure offering temporal scalability and low latency. This figure has been published in the SVC overview paper by Wiegand et al. [59] and is reproduced here with the kind permission of the authors.*

Although temporal scalability is well developed in SVC, an efficient access to random frames is not possible without highly reducing the compression performances. The same conclusion holds regarding spatial random access. This is due to the dependencies inherently associated to closed-loop prediction mechanisms. In this thesis, we investigate the possibility to achieve good coding efficiency without relying on the closed-loop prediction coding paradigm. Therefore, we investigate the conditional replenishment and parity-based refreshment mechanisms, shortly reviewed in the two following sections.

## 2.5 Conditional replenishment

Conditional replenishment has been introduced by Mounts [49] in 1969, in the early years of digital video coding. The basic concept is illustrated in Figure 2.9. Frames are divided in spatial zones which are encoded independently, and only the parts of the current frame that differ significantly from a reference maintained at the receiver are transmitted.

Compared with pixel-to-pixel 1-D DPCM, the most popular coding technique at the time, conditional replenishment technique is more efficient due to its ability to exploit inter-frame redundancy on blocks. About ten years later, in order to improve compression performances, conditional replenishment has been combined with an intraframe transform, using a two-dimensional variable rate Hadamard transform coder [33].

More recently, conditional replenishment has been exploited in several papers attracted by its INTRA coding scheme. For example, [46] and [47] integrate the conditional replenishment paradigm where it is used as a simple video coding method offering good flexibility for multicast distribution.

In this work, we have extended the original conditional replenishment framework by increasing the number of references candidates to approximate the spatial zones to transmit. Moreover, different coding techniques are considered to replenish these zones and the rate-distortion optimal technique is selected on the fly among a pre-computed set of replenishment option (see Section 3.4). This is illustrated in Figure 2.10. Depending on the content to transmit, the previous replenishment decisions and the available bit-budget, the system will decide to use a reference, to transmit fresh replenishment data, in a rate-distortion optimal way.

As the decoding of fresh data at the client side is achieved independently from the previously decoded frames, conditional replenishment can be considered as an open-loop coding mechanism. As long as replenishment decisions can be adapted on the fly, based on individual user requirements, random access to individual frames is possible and encoder[1] and decoder desynchronization has a much smaller impact on the system performances than for algorithms based on closed-loop prediction.

[1] In that case, the reference is simply unavailable and only fresh information is used for replenishment.

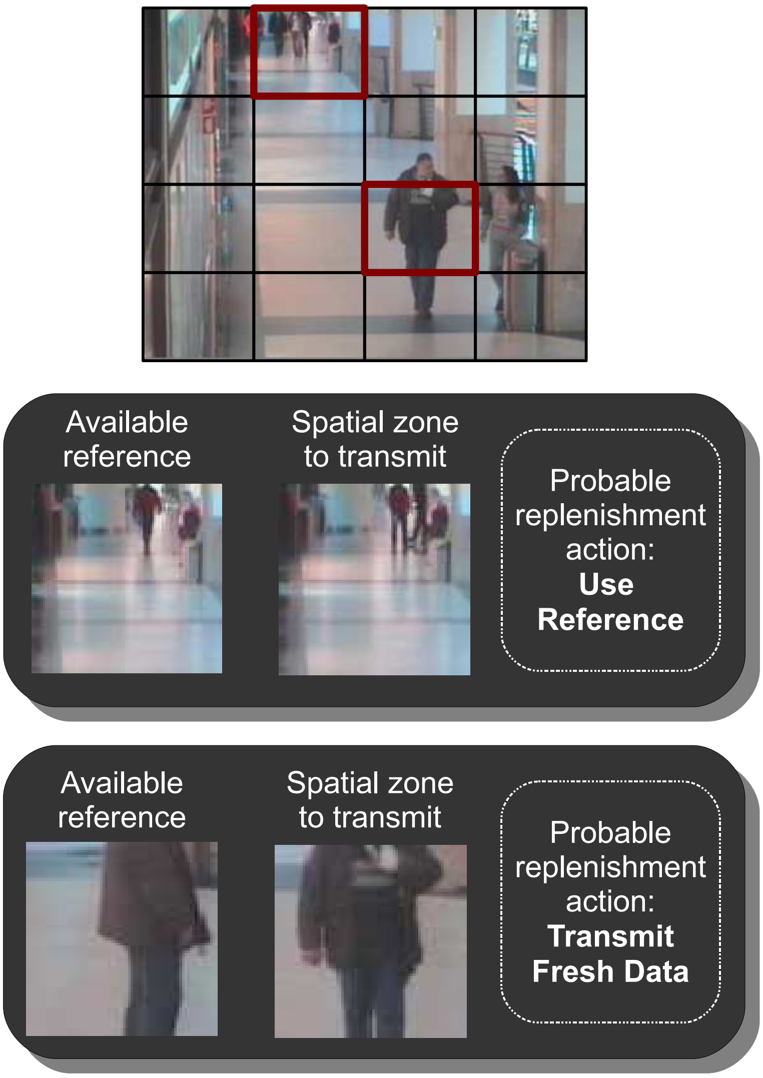

Available reference
Spatial zone to transmit
Probable replenishment action: **Use Reference**
Available reference
Spatial zone to transmit
Probable replenishment action: **Transmit Fresh Data**

Figure 2.9: *In the conditional replenishment framework, frames to transmit are divided in spatial zones which are either approximated by a reference available at the decoder, or replenished by fresh data. In this example, the reference will probably be used to approximate the first zone, as they are similar. In the second case, fresh data will probably be transmitted to replenish the second zone, since it significantly differs from the second reference.*

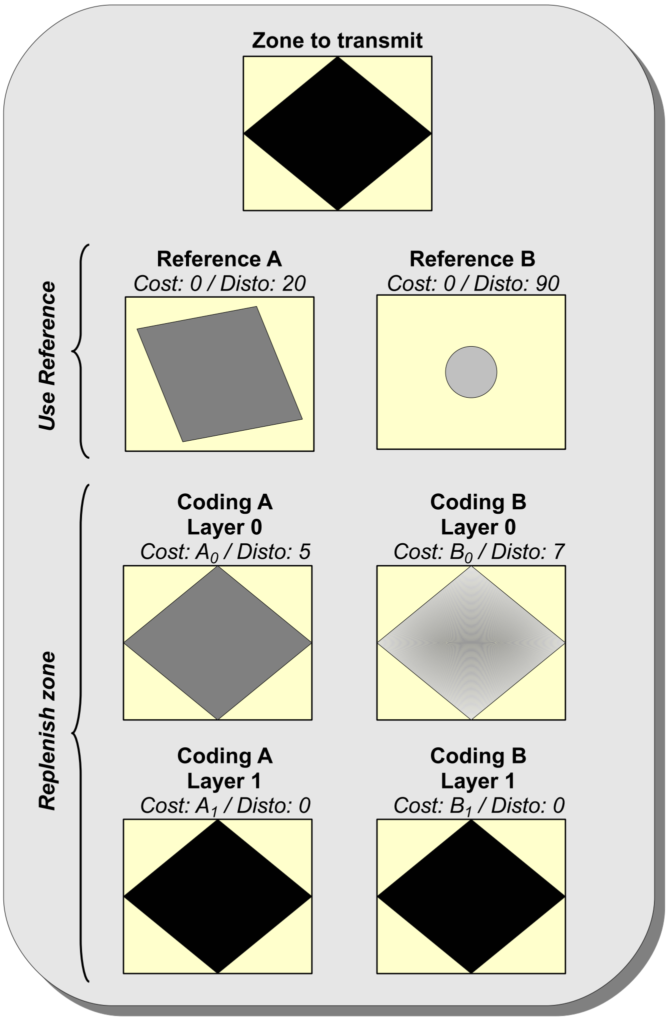


Figure 2.10: *Conditional replenishment with multiple references and replenishment techniques. In this example, two references and two coding techniques are available. A cost in bytes and a measure of the distortion is associated to each replenishment option. Two quality layers are generated by each coding technique, each low layer resulting in a lower cost and lower quality than the corresponding high layer. Depending on the available bit-budget, the system will approximate the zone to transmit with a reference or decide to replenish the zone by transmitting compressed data generated by one of the coding techniques, at a given quality level.*

## 2.6 Video coding with side information

In this section, we explain how the principles of coding with side information have been recently extended to video coding. It results in a codec that shifts the complexity associated with efficient temporal redundancy exploitation from the encoder to the decoder. More importantly, we will see that it also provides a solution to relax the synchronization constraint between the encoder and decoder, thereby alleviating the main drawback of strict closed-loop prediction systems.

Video coding with side information is based on the Slepian-Wolf theorem [63] published in 1973. Let us consider two statistically dependent signals $X$ and $Y$ respectively characterized by a transmission rate of $R_X$ and $R_Y$. The minimum lossless rate at which a signal $X$ can be transmitted is the signal entropy $H(X)$. By encoding both signals $X$ and $Y$ *together*, it is possible to reach a minimum lossless transmission rate of $H(X,Y)$, their joint entropy. Slepian and Wolf have shown that the same asymptotic performance is also achievable when the signals $X$ and $Y$ are encoded *separately*, as long as the two coded streams are decoded *jointly* and the following conditions are met [63]:

$$R_X \geq H(X|Y) \tag{2.2}$$

$$R_Y \geq H(Y|X) \tag{2.3}$$

$$R_X + R_Y \geq H(X,Y) \tag{2.4}$$

Figure 2.11 illustrates graphically these conditions. A similar result holds in the case of lossy coding and has been demonstrated in 1976 by Wyner and Ziv [68].

Video coding with side information focuses on one instance of the Slepian-Wolf and Wyner-Ziv theorems in which $Y$ is coded losslessly at rate $H(Y)$. In this case, it results from the theorem that $X$ can be coded at rate $H(X|Y)$, and recovered at the receiver with vanishing error probability. In this video context, $Y$ is considered as a reference frame stored at the client side and $X$ the frame to transmit. The reference frame $Y$ is usually the last decoded frame and is considered as a *side information*. The outcome of the Slepian-Wolf theorem in this context is obvious: with a sufficient knowledge of the correlation between $X$ and $Y$, the frame $X$ can be transmitted at a much lower rate, thanks to the exploitation by the decoder of the side information $Y$.

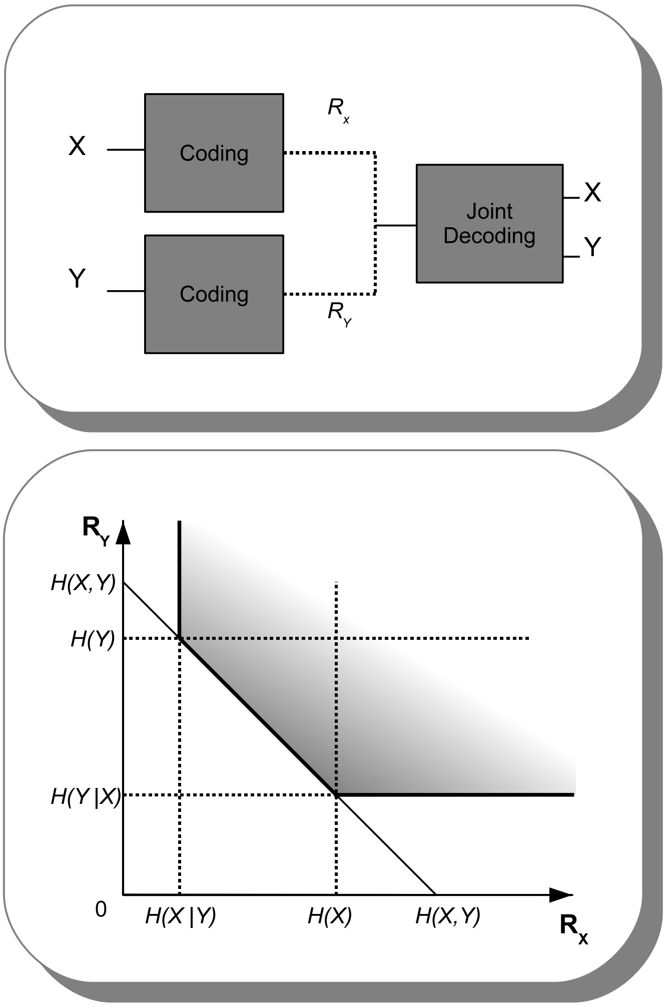


Figure 2.11: *Considering two correlated signals $X$ and $Y$ encoded independently, the Slepian-Wolf theorem defines the admissible rate region (shaded region below) for which a joint decoding of these signals is possible.*

We learn from the analysis of the conditional entropy $H(X|Y) = H(X) - I(X;Y)$ that two factors enable to decrease the rate at which we can transmit $X$:

- *$I(X;Y)$, the mutual information between $X$ and $Y$.* This value will be high if $X$ can be efficiently predicted from $Y$. This can be done by exploiting the temporal correlation between the reference and the image to transmit.

- *$H(X)$, the entropy of $X$.* In practice, the frame $X$ is encoded based on codewords that are shorter than the frame size. Encoding those codewords independently most often result in a significant increase of entropy, compared to $H(X)$. Hence, it is important to exploit the correlation between the codewords of $X$ so as to maintain the entropy of the actual codewords close to the initial frame entropy $H(X)$[1].

The proofs of the Slepian-Wolf and Wyner-Ziv theorems are asymptotical and non-constructive, and more than 30 years have passed before the first practical implementation of the theorem in a video coding context saw the light. In 2002, different ways of implementing such coding systems have been proposed [5, 56]. In these frameworks, the side information $Y$ is considered as a noisy version of $X$ and techniques coming from the conventional channel coding field are used to correct the side information. At the encoder, *parity bits* typically generated with Turbo Codes or LDPC Codes are calculated for $X$ and transmitted to the decoder where they are used to correct $Y$. The correlation between $X$ and $Y$ is often associated to a *Virtual Channel*, since the parity bits aim at correcting the errors introduced by this channel through which $X$ is sent and $Y$ is received. This process is depicted in Figure 2.12. Practically, the parity bits are computed based on the binary representation of the image coefficients of the frame to transmit, as depicted in Figure 2.13.

Throughout the years, the video coding system with side information approach has been improved in several ways, e.g. by adding a DCT transform to the encoding process [6], at the expense of an increased coding complexity. Beside low encoder complexity, video coding with side information has also been shown to be more robust to transmission errors [57].

As explained in the introduction chapter, our replenishment system makes use of parity bits to correct a reference considered as a side information. Parity bits are introduced to relax the closed-loop prediction constraint. Indeed, although a reference is exploited by the parity bits at the decoder, the decoding process fundamentally relies on statistical distribution inferred from the reference, and not on the reference itself. Specifically, parity bits complete the information provided by those distributions. Hence, the system is robust to some modifications of the reference.

[1] In our work, this is achieved by representing $X$ through spatially localized subband samples and by exploiting the frequency and spatial correlation between those samples.

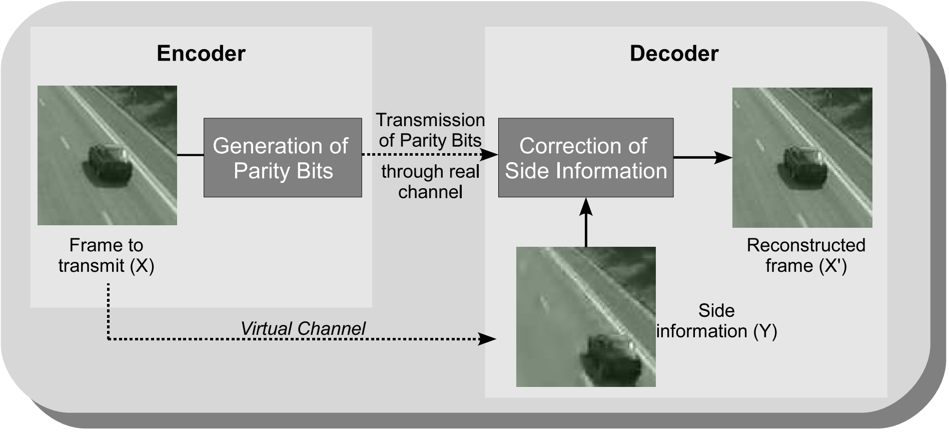
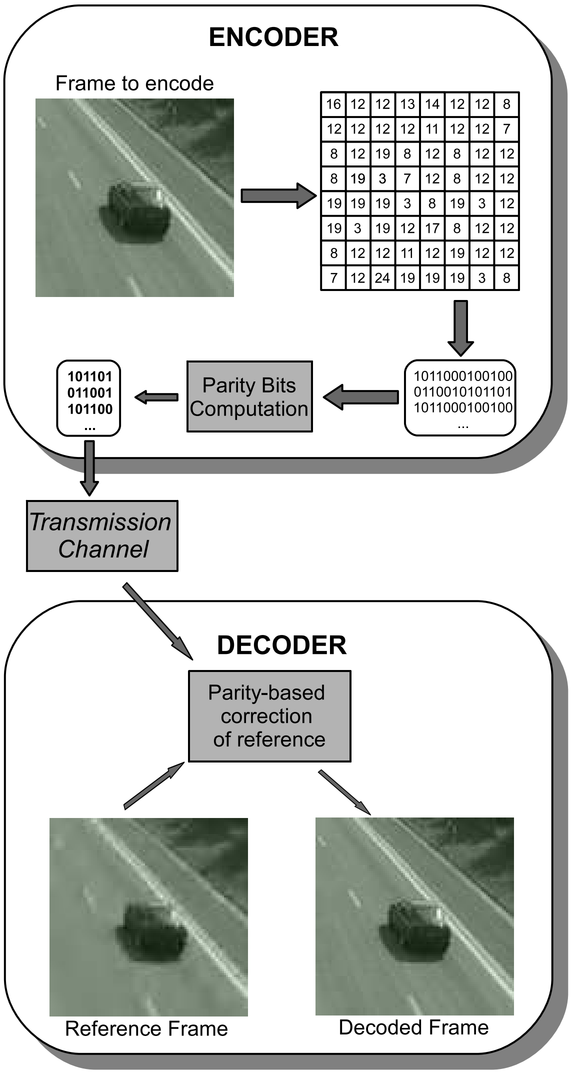


Figure 2.12: *Video coding system with side information. Parity bits are generated at the encoder based on the frame to transmit. With these parity bits, the decoder corrects the side information, which is usually the previous decoded frame, and generates the reconstructed frame.*

The main difference with previous parity-based video coding systems lies in the fact that coding with side information is combined with alternative open-loop coding options in a RD optimal way. Hence, we do not try to reduce the encoder complexity (at the cost of higher decoding complexity), but rather investigate whether parity bits can improve traditional the conditional replenishment methods which consist in transmitting fresh data or using the available reference.



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 16 | 12 | 12 | 13 | 14 | 12 | 12 | 8 |
| 12 | 12 | 12 | 12 | 11 | 12 | 12 | 7 |
| 8 | 12 | 19 | 8 | 12 | 8 | 12 | 12 |
| 8 | 19 | 3 | 7 | 12 | 8 | 12 | 12 |
| 19 | 19 | 19 | 3 | 8 | 19 | 3 | 12 |
| 19 | 3 | 19 | 12 | 17 | 8 | 12 | 12 |
| 8 | 12 | 12 | 11 | 12 | 19 | 12 | 12 |
| 7 | 12 | 24 | 19 | 19 | 19 | 3 | 8 |

Figure 2.13: *Practical method to compute the parity bits, illustrated in an encoding, transmission and decoding scheme. The coefficients of the frame to transmit are represented in their binary form. The parity bits are then computed based on this sequence of bits using channel codes like LDPC or Turbo Codes. At the decoder side, the binary representation of the reference frame is corrected with the help of the received parity bits, generating the decoded frame.*

## 2.7 Proposed Framework

The video system proposed in this work and depicted in Figure 2.14 integrates JPEG 2000 and parity coding within the conditional replenishment framework.

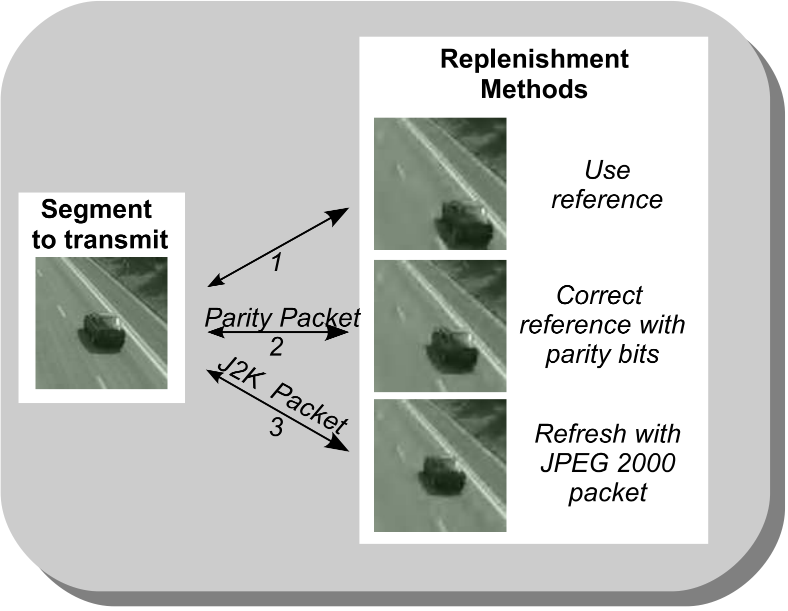


Figure 2.14: *Proposed replenishment methods. Given an image segment to transmit, the system recommends the client (1) to use the corresponding segment in the previously reconstructed frame considered as the reference, (2) to correct this reference using parity bits or (3) to refresh this reference by decoding the transmitted JPEG 2000 packet.*

As we will see further on in this work (Section 3.7), these two coding methods are complementary. JPEG 2000 is more efficient to replenish zones which significantly differ from the reference while parity bits are useful in intermediary situations in which the differences with the reference are not significant. This is illustrated in Figure 2.15.

The replenishment framework is presented in details in Chapter 3. Chapter 4 is devoted to the concept of parity based replenishment and finally, Chapter 5 presents how the proposed framework can be implemented in a low complexity scalable video server.

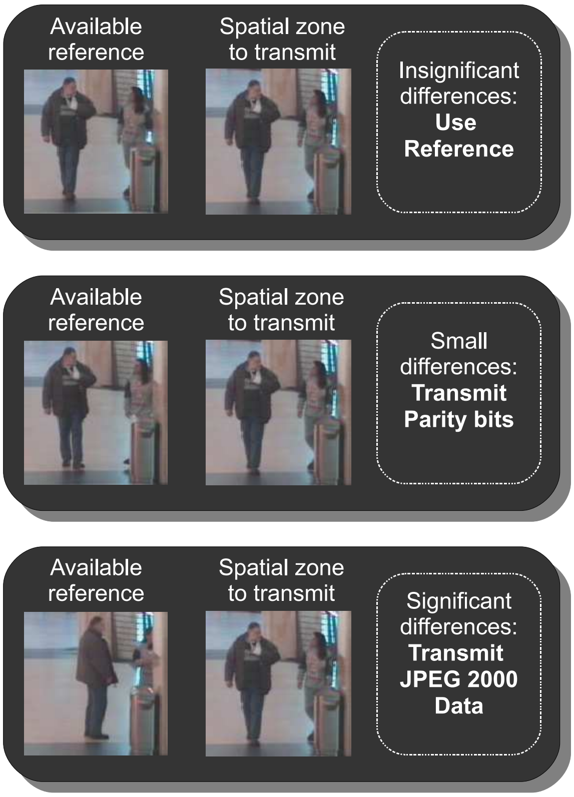


Figure 2.15: *The proposed conditional replenishment framework integrates two complementary coding methods. The choice between these methods depends on the magnitude of the differences between the zone to transmit and the available reference. It appears in Chapter 4 that with small differences, the transmission of parity bits is the best solution in a rate-distortion optimal framework. With higher differences, JPEG 2000 is the most efficient coding method.*

## 2.8 Conclusion

In this chapter, we have first presented the JPEG 2000 standard which offers a high scalability and compression efficiency for still images. Then, we have proposed an overview of three video coding techniques which differ in the way they exploit the temporal prediction. SVC, the most efficient solution in terms of compression performances, is based on a closed-loop prediction which requires a high constraint on encoder-decoder synchronization.

Video coding with side information relaxes this constraint by transmitting parity bit information, which rely on coherent reference statistics rather than on deterministic reference values. Finally, in the conventional conditional replenishment mechanism, the decoding of received INTRA packets is independent of the reference, opening the prediction loop and thereby reducing the encoder-decoder synchronization constraint. In practice, those INTRA packets are encoded based on the JPEG 2000 standard, which offers a high scalability and compression efficiency for still images.

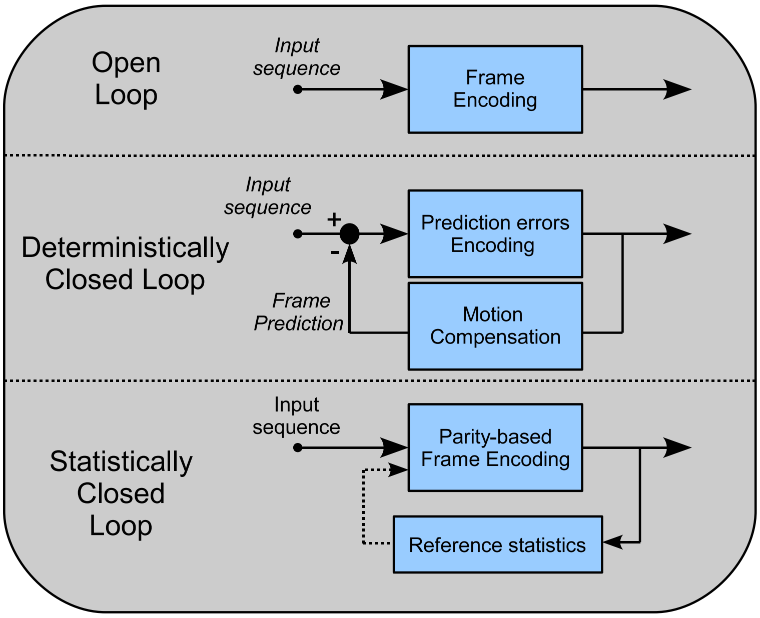


Figure 2.16: *Summary of the prediction loop models presented in this chapter. Compared to open loop, closed loop exploits the sequence temporal redundancy but requires a tight synchronization between encoder and decoder, thereby preventing an efficient random access to frames and increasing the sensitiveness to transmission errors. The proposed system is characterized by a partially closed-loop, which relaxes the synchronization constraint, offers an INTRA access to the compressed content while exploiting the sequence temporal correlation to improve the coding efficiency.*

Chapter 3 and 4 present our replenishment framework. They respectively integrate INTRA and parity based refreshment mechanisms to offer a highly scalable video coding solution that avoids closed-loop prediction, as depicted in Figure 2.16, while exploiting the sequence temporal correlation.

The relaxation of the closed-loop constraint permits to serve heterogeneous clients based on appropriate scheduling of pre-encoded sets of parity and JPEG 2000 packets. This flexibility is illustrated in Chapter 5.

# Scalable video streaming based on conditional replenishment

# 3

*Most parts of this chapter have been published in [20, 21].*

*This chapter presents the core architecture of our conditional replenishment system. Its main motivations, which are scalability and user-driven access to video content, are presented through a comparison of conventional codecs in several transmission scenarios. This comparison demonstrates that the replenishment framework offers good performance to serve very heterogeneous clients with a single pre-encoded content.*

*The proposed replenishment mechanism integrates two refreshment methods, JPEG 2000 and parity-bit coding, and can handle multiple references. To schedule these replenishment options, an optimal rate-distortion allocation process takes into account network conditions and individual user preferences about regions of interest within the browsed content. Based on pre-computed rate-distortion values and low complexity operations, this adaptation can be achieved in real-time, during the transmission to multiple heterogeneous users.*

## 3.1 Introduction

Our work targets applications requiring a highly scalable access to stored video sequences. Users logging to the system are expected to have very different profiles, resources and interests in the content. They are connected with asynchronous clients to a unicast network. A scalable compression technique is envisioned to avoid multiple versions of the same content and reduce memory requirements at the server. Common actions on the potentially high resolution video sequences are zooming, cropping and extracting low-resolution versions of both consecutive or individual frames. Hence, scalability in resolution, quality and spatial access is required, as well as random access capabilities to individual frames. In order to reduce bandwidth requirements, an efficient compression of the sequences is also necessary. To answer both requirements of scalability and compression, we have chosen to encode the sequences with the JPEG 2000 compression algorithm which provides a fine grained spatial and temporal scalability.

As temporal redundancy is not exploited due to the INTRA nature of JPEG 2000 video compression, the system performances are significantly penalized compared to INTER compression schemes, such as MPEG [9, 44, 64]. In order to circumvent this drawback, we have adapted conditional replenishment principles to the specificities of JPEG 2000 and have introduced a background estimate as an alternative replenishment option. This has been shown to be particularly useful in video-surveillance scenarios dealing with still cameras. To further improve the performance of the system, a new replenishment method based on the parity correction of the side information available at the client side has been proposed. In Chapter 5, this parity-based solution is also shown to be robust to reasonable desynchronization between encoder and decoder, thereby mitigating the main drawback of closed-loop video codecs when addressing multiple heterogeneous users and error-prone channels.

In a conditional replenishment framework, each frame to transmit is divided into several zones which are replenished independently. In our work, these zones correspond to blocks of wavelet coefficients and can be updated in three ways. First, the client can replenish the zone based on a reference frame, which typically corresponds to the last decoded frame or to a background estimate. The second solution consists in transmitting parity bits to correct the reference at the client side. Finally, the third solution consists in the transmission of a JPEG 2000 packet to fill in the zone with fresh information.

Besides, the proposed framework enables the rate-distortion allocation process to take into account individual user preferences about regions of interest within the browsed content. This knowledge is exploited independently of the compression stage, which means that it can be provided a posteriori, at transmission time, by each individual user. In a sense, this semantic adaptation of the compressed content can be considered as an additional dimension of the notion of scalability.

Our work presents similarities with [12] which also focuses on a coding framework able to handle uncertainty on the prediction status at the decoder. However, our work brings two major contributions compared to existing research. First, our framework offers an extremely rich scalability to the user. Second, we propose an alternative to the transmission of parity bits by offering JPEG 2000 replenishments, which significantly improves the system performances (see Section 3.7.1).

In the remainder of the chapter, we motivate our work in Section 3.2, through a comparison of several video codecs in four typical browsing scenarios. The wavelet image representation offering the desired high scalability and the associated distortion metrics considered in this work are presented in Section 3.3. The scheduling strategy enabling a rate-distortion optimal replenishment of visualized content is presented in Section 3.4. Section 3.5 and 3.6 respectively present the background estimation module and the global overview of the system architecture. Its performances are presented and discussed in Section 3.7. Finally, Section 3.8 concludes this chapter.

## 3.2 Motivation: Remote interactive browsing in a surveillance context

In order to motivate the use of JPEG 2000 for storing and disseminating surveillance video content, it is interesting to consider a typical interactive video browsing scenario and to compare the channel and computational resources required when accessing remotely pre-recorded content based either on hybrid (INTER) or JPEG 2000 (INTRA) compression formats.

In the envisioned scenario, a graphical user interface (GUI) enables the human controller to visualize the chronology of recorded - and possibly pre-analyzed - events through a timeline of low-resolution key frames (scenario 1). The user can then select some time segments of the video to display at higher resolution (scenario 2). (S)he can also interactively select and further zoom in on some areas of interest, in a particular video segment (scenario 3) or frame (scenario 4) of the displayed scene.

Table 3.1 considers a content captured at 15 fps with a still 2 Mpixels camera and reviews the four access scenarios involved in the browsing ses-

| | Scenario | Encoded signal resolution | Displayed fraction of initial image |
|---|---|---|---|
| 1 | Time-line of very-low-resolution frames | 192 × 144 | 1/1 |
| 2 | Low-resolution video segment | 384 × 288 | 1/1 |
| 3 | Zoom in (spatially) random video segment | 768 × 566 | 1/4 |
| 4 | Zoom$^+$ in (spatio-temporally) random frame segment | 1536 × 1132 | 1/16 |

Table 3.1: *Content access scenarios definition. Content has been captured at 15 fps, with a 2 Mpixels camera.*

sion described above. The scenarios differentiate themselves by the spatial resolution at which they access the content, and by the particular duration of the video segment they actually access. In particular, scenario 1 envisions the display of a chronological time-line of very low resolution frames. Scenario 2 considers the display of a video segment at low resolution. Scenario 3 considers a cropped and subsampled version of the video, while scenario 4 considers the access to a 384x288 window in a randomly selected frame of the original video sequence.

| Scenario | J2K | **JPRB** | AVC (I+14P) | AVC (All I) | SVC | AVC FMO (I+14P) |
|---|---|---|---|---|---|---|
| 1 (Frames: kbits/sample) | 24 | **24** | 20 | 20 | 20 | 20 |
| 2 (Video: kbits/sec) | 1020 | **189** | 78 | 840 | 93 | 78 |
| 3 (Video: kbits/sec) | 702 | **148** | 215 | 2190 | 251 | 101 |
| 4 (Frames: kbits/sample) | 32 | **32** | 494 | 415 | 537 | 57 |

Table 3.2: *Average bandwidth consumption for each access scenario and for distinct encoding schemes, with a PSNR of* 35 dB*. For the J2K and the proposed JPRB methods, a single fine-grained codestream is generated for the four scenarios and could be used to meet other rate constraints. SVC and AVC streams are generated to target the four pre-defined scenarios, and different versions of the AVC stream are generated for each scenario while SVC only requires a single stream. As the first and last scenarios are related to the transmission of arbitrary frames, the bandwidth consumption is measured in kbits/frame. For the other scenarios, kbits/sec measure the required bitrate.*

For each scenario, Table 3.2 compares the average bitrate required to access a typical surveillance content based on four distinct codecs. J2K (column 1) encodes and decodes the video images based on the JPEG 2000 algorithm. JPRB (JPEG 2000 and Parity Replenishment with Background - column 2) refers to the original solution described and validated along this chapter. In short, it relies on both JPEG 2000 and parity-bit packets and implements multiple-reference and RD optimized conditional replenishment mechanisms to reduce the bandwidth consumption when accessing video segments characterized by stationary backgrounds. The two next solutions build on the H.264/AVC standard, and encode one INTRA frame every second (column 3) or all frames in INTRA (column 4). For both AVC solutions, four distinct streams are generated, corresponding to the four spatial resolutions considered by the scenarios in Table 3.1. The two last solutions respectively built on SVC and AVC FMO are detailed below. For each coding scheme and each spatial resolution, the encoding parameters have been tuned to reach an approximate PSNR of 35 dB, offering an equitable visual quality for all scenarios.

In Table 3.2, bandwidth is defined in kbits/sample or kbits/sec depending on whether the scenario considers the access to an individual frame or to a video segment lasting several seconds. As AVC is not supposed to provide spatio-temporal random access capabilities, we assume that entire frames have to be decoded to access the frame/video segment of interest in scenarios 3 and 4. Moreover, partial GOPs have to be decoded to access a single and randomly selected frame with AVC I+14P in scenario 4. Hence, depending on the position of the frame to access in the GOP, a number of P frames have to be decoded in addition to the first Intra frame of the GOP. This explains why the cost to access a sample in scenario 4 is higher for AVC I+14P than for AVC I.

A careful analysis of Table 3.2 reveals that the INTRA nature of JPEG 2000 strongly penalizes J2K compared to AVC (I+14P) when video segments have to be transmitted. It also reveals that J2K provides an attractive solution when random spatial and/or temporal access is desired (scenarios 1 and 3) or when a single frame has to be displayed (scenario 1 and 4). The lack of spatio(-temporal) random access capabilities significantly penalizes AVC-based solutions compared to J2K and JPRB solutions in scenarios 3 and 4. Interestingly, we observe that our proposed JPRB solution preserves the advantages of J2K, while smoothing out its main drawback. Specifically, JPRB appears to be the only solution capable of dealing with all scenarios with a bandwidth of 200 kbps and a latency smaller than one second for scenario 4. This fact definitely demonstrates the relevance of our study. A summary of this comparison is proposed in Table 3.3.

| Codec | Compression efficiency | Scalability |
|---|---|---|
| J2K | Low | High |
| **Proposed JPRB** | **High** | **High** |
| AVC (I + 14 P) | High | Low |
| AVC (All I) | Low | Medium |

Table 3.3: *Summary of codecs comparison in terms of compression efficiency and scalability for the four discussed scenarios.*

Here, it is worth noting that our system does not implement any motion compensation algorithm. Hence, it is dedicated to scenarios characterized by a stationary background, as encountered in video-surveillance context. The purpose of the thesis was to validate the parity-based replenishment framework, and the extension to any kind of moving images should be considered as part of future work[1].

Before moving to the actual description of our replenishment solution, it is worth making two comments about AVC-based video coding schemes.

First, the scalable extension of MPEG-4 AVC, namely SVC [50] which was presented in Section 2.4, enables the encoding of a high-quality video bitstream that contains one or more subset bitstreams that can themselves be decoded with a complexity and reconstruction quality similar to that achieved using MPEG-4 AVC with the same amount of data as in the subset bitstream. Hence, SVC prevents the multiplication of streams, but does not fundamentally affect the conclusions drawn from Table 3.2. This is illustrated by the column 5 (SVC) in Table 3.2. There we present a SVC solution for which the first resolution has been encoded based on a I + 14 P GOP structure. For the second and third resolutions, frames are predicted based on the highest lower resolution and the previous frame. To improve random access capabilities, the last and finest resolution only exploits the lower resolution as a reference (and not the previous frame). We observe in Table 3.2 that SVC achieves about the same performance as the four versions envisioned for AVC I + 14P. This is not surprising since SVC encounters some (minor) penalty when embedding the four versions

[1] We will see in Chapter 4 that the specificities of the coding system justify a careful and dedicated design of the motion compensation engine.

in a single bitstream.

Secondly, it is possible to exploit the flexible macroblock ordering concept of MPEG-4 AVC to define a grid of block-shaped slices that can be accessed independently, thereby improving the spatially random access capabilities of AVC, at the expense of some coding efficiency[1]. The last column in Table 3.2 presents the bandwidth requirements corresponding to the four envisioned scenarios when the AVC I+14P codec considers independent slices of $64 \times 64$ pixels. We conclude that, for a given targeted quality or bit budget, results equivalent or even slightly better than the one obtained with J2K could be obtained with MPEG-4 AVC or SVC standards for the fourth scenario, by encoding high resolution frames in INTRA (to allow for random temporal access) and based on a set of independent slices. However, in such scenario, JPEG 2000-based solutions still remain attractive due to their inherent fine grained embedded nature. With JPEG 2000, there is no need to work with sophisticated decoder architectures, able to handle a discrete set of (embedded) versions of the same content, encoded at distinct quality and resolution levels. With JPEG 2000, the client simply handles and decodes conventional JPEG 2000 packets to browse arbitrary portions of the content in a progressive and fine grained manner, both in quality and resolution. Such progressivity is especially desired when serving heterogeneous terminals, for which transmission resources and interest in the scene are defined by each individual user at transmission time.

Hence, the core of this chapter mainly consists in explaining and demonstrating how dedicated conditional replenishment mechanisms efficiently preserve the fine grained flexible nature of JPEG 2000 image representation, to adapt streamed content to individual user needs while saving some bit budget, when serving surveillance video segments, thereby reaching the performance presented in the JPRB column of Table 3.2.

## 3.3 Image representation and distortion metrics for JPEG 2000 compliant replenishment

This section explains how the adoption of the JPEG 2000 image representation, combined with the proposed replenishment mechanisms, offers

[1] For example, [45] considers a low resolution base layer encoded with motion compensation, and a high resolution enhancement layer encoded in a set of independent slices that are only predicted from the base layer.

temporal, spatial, quality and resolution scalability, as well as direct integration of user preferences at transmission time.

### 3.3.1 Image representation

The JPEG 2000 standard [3], which has been presented in Section 2.3, describes images in terms of their discrete wavelet coefficients. The zones considered for conditional replenishment have to coincide to precincts[1] so that the replenishment by fresh data simply corresponds to the selection and transmission of appropriate JPEG 2000 packets.

The JPEG 2000 subdivision of wavelets resolutions into precincts also conditions the generation of parity bits. Specifically, each parity packet contains the parity bits correcting the precinct coefficients at a given quantization level defined for each code-block. In this work, the parity quantization levels of each code-block have been set to the same value as the JPEG 2000 quantization levels for simplicity[2].

The way the reference is corrected with the parity bits and the way these parity bits are generated is out of the scope of this chapter and will be detailed in Chapter 4. At this point, the only thing we need to know is that parity bits offer a set of alternative rate-distortion trade-offs for the zone to replenish.

In summary, the wavelet transform and the independent coding of precincts support the spatial and resolution scalability of our system. The bit-plane description of coefficients combined with a layering approach offers the quality scalability. Figure 3.1 illustrates these three scalabilities.

In addition to these scalabilities, the proposed video system offers temporal scalability and a random access to the frames, since the sequences are encoded in INTRA with the JPEG 2000 format.

### 3.3.2 Distortion metrics

In our work, the distortion metric is computed based on the Square Error (SE) of wavelet coefficients, and approximates the reconstructed image

[1] As explained in Section 2.3, precincts are spatial subdivisions of the wavelet resolutions defined by the JPEG 2000 standard.

[2] Since these quantization levels have been calculated by minimizing the Lagrangian cost function $\mathcal{L}(\lambda)$ defined in Equation 2.1 (page 13) using the distortion and rate issued from the JPEG 2000 coding and not the parity coding, the parity truncation points are not necessarily RD optimal like the JPEG 2000 ones [67].

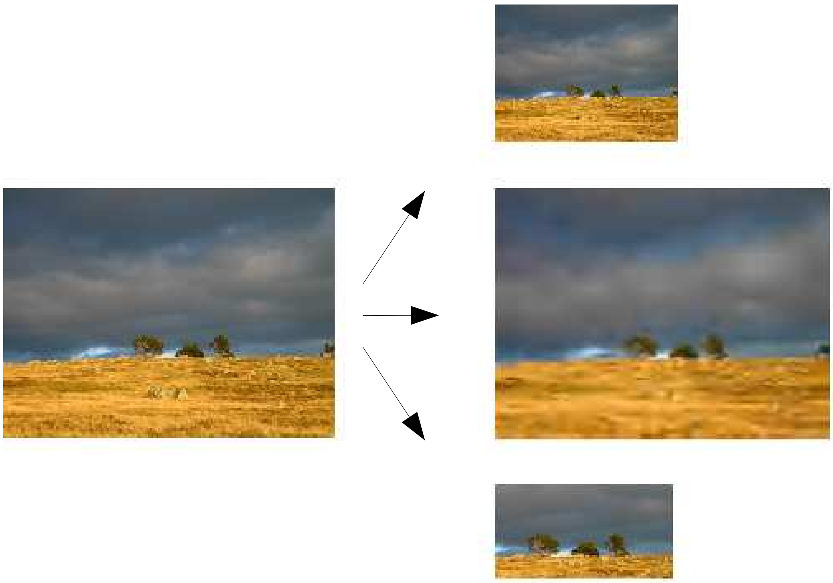


Figure 3.1: *The image representation offers resolution (top), quality (middle) and spatial (low) scalability.*

square error [67]. Formally, let $\mathcal{B}_i$ denote the set of code-blocks associated to precinct $i$, and let $c_b[k]$ and $\hat{c}_b[k]$ respectively denote the two-dimensional sequences of original and approximated subband samples in code-block $b \in \mathcal{B}_i$. The distortion $d(i)$ associated to the approximation of the $i^{th}$ precinct is then defined by

$$d(i) = \sum_{b \in \mathcal{B}_i} \gamma_b^2 \sum_{k \in b} (\hat{c}_b[k] - c_b[k])^2 \tag{3.1}$$

where $\gamma_b$ denotes the L2-norm of the wavelet basis functions for the subband to which code-block $b$ belongs [67]. The $\gamma$ values of the 5x3 and 9x7 discrete wavelet transforms considered in this work can be found in the JPEG 2000 standard [3].

### 3.3.3 Semantical weighting of the distortion

As an alternative to the conventional SE metric presented above, a different distortion can be considered, based on semantically meaningful weighting of the SE so as to take into account the a priori knowledge one might have about the semantic significance of approximation errors.

Assuming that the information about the semantic relevance of approximation errors is provided at the precinct level, we define $d'(i)$, the semantically weighted distortion to be

$$d'(i) = w(i)d(i) \tag{3.2}$$

where $w(i)$ denotes the semantic weight assigned to the $i^{th}$ precinct and $d(i)$ is the distortion defined in Equation 3.1.

This is a key difference with earlier contributions that have considered semantically meaningful weighted distortion metrics in the past. In this field, it is worth to mention [11, 26] in which the segmentation is directly integrated within a complete region-based coding scheme and [4] where semantic analysis and the corresponding content annotations are exploited for object-based encoders, such as MPEG-4 [37], as well as for frame-based encoders, such as MPEG-1.

In these contributions, the metrics are exploited either before or during the encoding step. In contrast, our work supports the posterior definition of semantic weights given the pre-encoded stream, at transmission time for each client, thereby allowing to serve multiple clients, with different semantic interests, based on a single JPEG 2000 codestream. An example of transmission session influenced by such semantical weighting of the content is presented in Section 3.7.3, page 61.

From a functional point of view, using the weighted distortion instead of conventional distortions does not lead to any significant increase of computational complexity . In particular, it is worth noting that the convex-hull analysis performed on non-weighted distortions, as presented in the next section remains valid as long as the weighting affects in a similar way all the packets of a precinct, which is the case if weights are defined at the precinct level.

## 3.4 Rate-distortion optimal replenishment

Considering the above described wavelet based image representation, we now explain how to select the JPEG 2000 and parity packets of the current image codestream so as to maximize the reconstructed image quality, given a targeted transmission budget and a reference image available at the receiver.

As the JPEG 2000 and parity codestreams are made of a set of precincts organized in a hierarchy of layers (see Section 2.3.3), the problem consists in selecting for each precinct the type of replenishment (reference, parity or JPEG 2000) and its quality level, so as to maximize the reconstructed quality (or minimize the distortion) under the bit budget constraint.

To solve the problem efficiently, we assume the additive distortion metric presented in Sections 3.3.2 and 3.3.3, for which the contribution provided by multiple precincts to the entire image distortion is equal to the sum of the distortion computed for each individual precinct.

### 3.4.1 RD optimality using the convex-hull approximation

The problem of rate-distortion (RD) optimal allocation of a bit budget across a set of image blocks characterized by a discrete set of RD trade-offs has been extensively studied in the literature [53, 54, 61]. An excellent overview of the problem applied to image and video compression can be found in [55].

Under strict bit budget constraints, the problem is hard and relies on heuristic methods or dynamic programming approaches to be solved [53]. In contrast, when some relaxation of the rate constraint is allowed, Lagrangian optimization and convex-hull approximation can be considered to split the global optimization problem in a set of simple block-based local decision problems [54, 61]. The convex-hull approximation consists in restricting the eligible transmission options for each block (or precinct in our case) to the RD points sustaining the lower convex hull of the available RD pairs of the block. Global optimization at the image level is then obtained by allocating the available bit-budget among the individual code-block convex-hulls, in decreasing order of distortion reduction per unit of rate.

#### Problem definition

For a given frame to transmit, we assume that $N$ precincts have to be encoded using a given set $\mathcal{Q}$ of $M$ admissible replenishment solutions, such that the replenishment choice $x(i)$ for a precinct $i$ induces a distortion $d_{ix(i)}$ for a cost in bytes equal to $s_{ix(i)}$. The objective is then to find the allocation $\mathbf{x} \in \mathcal{Q}^N$ which assigns a replenishment choice $x(i)$ to precinct $i$, such that the total distortion is minimized for a given rate constraint.

In our case, the replenishment choice $x(i)$ refers to one of the $M$ references, JPEG 2000 or parity replenishments[1], $0 < x(i) \leq M$.

Formally, the rate-distortion optimal bit allocation problem is then formulated as follows:

**Optimal rate-constrained bit allocation** For a given target bit-budget $B_T$, find $\mathbf{x}^*$ such that

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_{i=1}^{N} d_{ix(i)} \tag{3.3}$$

subject to

$$\sum_{i=1}^{N} s_{ix(i)} < B_T. \tag{3.4}$$

### Lagrangian formulation and approximated solution

Strictly speaking, the above formulation corresponds to a Knapsack problem [25], which can be solved at high computational cost using dynamic programming [25, 39]. Hopefully, in most communication applications, the bit-budget constraint is somewhat elastic. Buffers absorb momentary rate fluctuations, so that the bits that are saved (overspent) on the current (fraction of) the image just slightly increment (decrement) the budget allocated to subsequent (fraction of) images, without really impairing the global performance of the communication.

Hence, we are interested in finding a solution to (3.3), subject to a constraint $B'$ that is reasonably close to $B_T$. This slight difference dramatically simplifies the RD optimal bit allocation problem, because it allows the application of the Lagrange-multiplier method. We now state the main and fundamental theorem associated with Lagrangian optimization, because it sustains our subsequent developments.

**Theorem 1.** *For any $\lambda \geq 0$, the solution $\mathbf{x}^*_\lambda$ to the unconstrained problem*

$$\mathbf{x}^*_\lambda = \arg\min_{\mathbf{x}} \sum_{i=1}^{N} d_{ix(i)} + \lambda \sum_{i=1}^{N} s_{ix(i)} \tag{3.5}$$

[1] Recall that both JPEG 2000 and parity replenishments can be achieved at different quality layers, each of these layers corresponding to a distinct replenishment solution.

*is also the solution to the constrained problem (3.3) with the constraint* $B_T = \sum_{i=1}^{N} s_{ix^*_\lambda(i)}$.

*Proof:* To simplify notations, we let $D(\mathbf{x})$ and $B(\mathbf{x})$ respectively denote $\sum_{i=1}^{N} d_{ix(i)}$ and $\sum_{i=1}^{N} s_{ix(i)}$.

For the solution $\mathbf{x}^*_\lambda$, we have $D(\mathbf{x}^*_\lambda) + \lambda B(\mathbf{x}^*_\lambda) \leq D(\mathbf{x}) + \lambda B(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{Q}^N$. Equivalently, we have $D(\mathbf{x}^*_\lambda) - D(\mathbf{x}) \leq \lambda(B(\mathbf{x}) - B(\mathbf{x}^*_\lambda))$, for all $\mathbf{x} \in \mathcal{Q}^N$. Hence, because $\lambda > 0$, for all $\mathbf{x} \in \mathcal{Q}^N : B(\mathbf{x}) \leq B(\mathbf{x}^*_\lambda)$, we have $D(\mathbf{x}^*_\lambda) - D(\mathbf{x}) \leq 0$. That is, $\mathbf{x}^*_\lambda$ is the solution to the constrained problem when $B_T = B(\mathbf{x}^*_\lambda)$. □

This theorem says that to every nonnegative $\lambda$, there is a corresponding constrained problem whose solution is identical to that of the unconstrained problem. As we sweep $\lambda$ from zero to infinity, sets of solutions $\mathbf{x}^*_\lambda$ and constraints $B(\mathbf{x}^*_\lambda)$ are created. Our purpose is thus to find the solution which corresponds to the constraint that is close to the target bit-budget $B_T$.

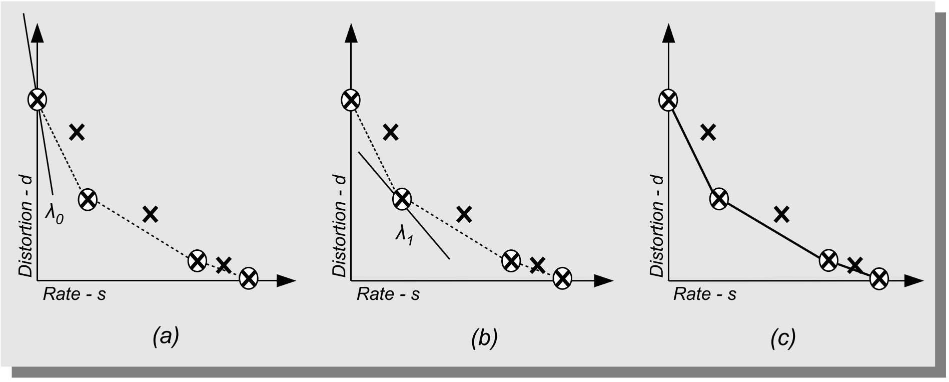


Figure 3.2: *Examples of Lagrangian-based bit allocation. In all graphs, the crosses depict possible operating points for a given code-block. Circled crosses correspond to RD convex-hull points, which provide the set of solutions to the unconstrained bit allocation problem. (a) and (b) depict the 'first hit' solution for two distinct values of* $\lambda$. *(c) plots the lower convex-hull.*

We now explain how to solve the unconstrained problem. For a given $\lambda$, the solution to (3.5) is obtained by minimizing each term of the sum separately. Hence, for all $i$,

$$x(i)^*_\lambda = \arg\min_{x(i)}(d_{ix(i)} + \lambda\, s_{ix(i)}) \tag{3.6}$$

Minimizing (3.6) intuitively corresponds to finding the operating point of the $i^{th}$ code-block that is "first hit" by a line of absolute slope $\lambda$ in a rate-distortion graph. See the examples in Figure 3.2. The convex-hull RD points are defined as the $(d_{ix(i)}, s_{ix(i)})$ pairs that sustain the lower convex-hull of the discrete set of operating points of the $i^{th}$ code-block. For simplicity, we re-label the $M_H(i) \leq M$ convex-hull points, and denote $(d^H_{ik}, s^H_{ik}), k \leq M_H(i)$ to be their rate-distortion coordinates. When sweeping the $\lambda$ value from infinity to zero, the solution to Equation 3.6 goes through the convex-hull points from left to right. Specifically, if we define $S_i(k) = (d^H_{ik} - d^H_{i(k+1)})/(s^H_{i(k+1)} - s^H_{ik})$ to be the slope of the convex-hull after the $k^{th}$ point, the $k^{th}$ point is optimal when $S_i(k-1) > \lambda > S_i(k)$, i.e. as long as the parameter $\lambda$ lies between the slopes of the convex-hull on both sides of the $k^{th}$ point.

In practice, the convex-hull approximation consists in restricting the eligible transmission options for each block (or precinct) to the RD points sustaining the lower convex hull of the available RD points of the block. In our case, this corresponds to the computation, for each precinct, of the convex-hull sustaining the JPEG 2000, parity and the reference RD points. This is depicted in Figure 3.3 where we observe that in this particular case, the only replenishment possibility at very low bitrates is to use the reference. With higher bitrates, the parity replenishment becomes more interesting than JPEG 2000 until a certain bitrate threshold.

At the image level, RD optimality is achieved by ensuring that each precinct selects its operating point, here replenishment solution, based on the same rate-distortion trade-off, as determined by the $\lambda$ parameter. The set of global solutions to the unconstrained problem is obtained by sweeping $\lambda$ from infinity to zero. While reducing the value of $\lambda$, the optimal solution to (3.6) progressively moves along the convex-hull for each precinct, ending up in choosing replenishment options with an increasing rate. The process naturally covers the entire set of solutions to the unconstrained problem, in increasing order of byte consumption and image reconstruction quality. Under a budget constraint $B_T$, we are interested in the solution that maximizes the quality while keeping the bit-budget below the constraint.

When the RD optimal point is reached, the optimal replenishment

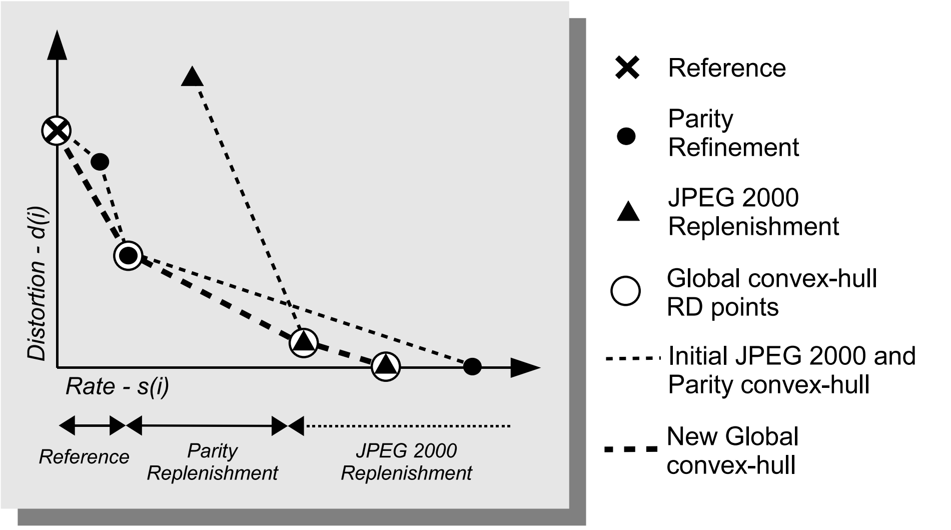


Figure 3.3: *Rate-distortion representation of the replenishment decisions for a given precinct. Depending on the available bitrate, the client will use the reference (cross), receive a parity packet (dot) or JPEG 2000 packets (triangles). The original JPEG 2000 RD points and the resulting replenishment decisions lie on convex-hulls.*

method corresponding to that $\lambda$ is selected for each precinct, and the other replenishment options are discarded.

It is noteworthy that this method, which is RD optimal at frame level, is applied in a greedy way for the video sequence, i.e. we consider a horizon limited to the current frame to encode for the sequence optimal allocation. Most video allocation processes are also characterized by such optimality at image level and sub-optimality at sequence level.

### Summary of precinct RD optimality

As a consequence of the above observations, overall RD optimality can be achieved at the image level by selecting the JPEG 2000 and parity packets so as to replenish the image precincts in decreasing order of benefit per unit of rate, up to exhaustion of the transmission budget [61]. This approach is inspired by the one defined in [15], but has been adapted to account for the availability of a reference image and two coding methods.

The solution is RD optimal in the sense that, for the achieved bit-budget, it is not possible to attain a lower reconstructed image distortion based on different allocation decisions. Indeed, by construction, it is not possible to find a non-selected replenishment option that provides a larger gain per unit of rate than the gain provided by already selected options.

### 3.4.2 Practical implementation of the RD optimal allocation process

Now that we have explained how RD optimality is achievable at image level, we describe how the practical RD optimal scheduling can be implemented. Formally, this iterative process can be defined as follows.

Let $o(i,m)$ denote the replenishment option already selected for the $i^{th}$ precinct at the iterative step $m$, $o^{+}(i,m)$ denote the following convex-hull optimal replenishment at step $m$ and $o^{ref}(i)$ denote the replenishment solution consisting in using the reference for the $i^{th}$ precinct. Based on these definitions, at the initial step, we have $o(i,1) = o^{ref}(i)\ \forall\, i$. Then, at each step $m$, the greedy process decides to improve the quality of precinct with index $i_m^*$ that provides the largest decrement in distortion per unit of transmission, i.e.

$$i_m* = \arg\max_{1\leq i\leq N} \frac{\left(d^{o(i,m)}(i) - d^{o^{+}(i,m)}(i)\right)}{\left(s^{o^{+}(i,m)}(i) - s^{o(i,m)}(i)\right)} \tag{3.7}$$

where $d^{o(i,m)}(i)$ and $s^{o(i,m)}(i)$ are respectively the distortion and cost in bytes of the replenishment option $o(i,m)$.

To prepare the next iteration, $o(i,m+1)$ is set to $o(i,m)\ \forall i \neq i_m^*$, and to $o^{+}(i_m^*,m)$ when $i = i_m^*$. The process goes on iterating on $m$ as long as the bit budget is not exhausted.

Practically, this scheduling process can be efficiently implemented by first pre-calculating the fraction in the right term of equation 3.7, which corresponds to the next replenishment option gain. The replenishment options are then selected by decreasing order of gain until exhausting the rate.

This possibility to integrate pre-computed values in the rate-distortion optimal allocation process is the key for the design of a low complexity server, as we will see in Chapter 5.

## 3.5 Background and foreground extraction of video-surveillance sequences

In this section, we present a method to extract a dynamic foreground and static background in a video-surveillance sequence. These two regions will be used in the experimental validation section to demonstrate the flexibility of the replenishment system. The background will serve as a second possible reference for the replenishment system and the foreground, considered as defining Regions of Interest (RoI) for the user, will guide the semantical weighting of the distortion metric during the allocation process.

### 3.5.1 Background estimation

The goal of the background estimation process is to create a background reference frame, which is considered as a second reference candidate for the replenishment, besides the previous decoded frame.

In practice, the background estimation is performed on a sliding window, and is based on a real-time statistical segmentation algorithm using a mixture of Gaussians modeling for the luminance of each pixel [65] [36] [69]. This approach automatically supports backgrounds with multiple states like blinking lights, grass and trees moving in the wind, acquisition noise, etc. Furthermore, the background model naturally updates in an unsupervised manner when the scene conditions are changing.

Figure 3.4 shows the mixture of Gaussians for one pixel at a given time. It aggregates all luminance values observed for that specific pixel in the previous frames belonging to the sliding window. The current pixel luminance is compared to the current mixture. We consider that it belongs to one of the Gaussians if the distance between the current pixel luminance and the Gaussian mean is lower than a given threshold proportional to the considered Gaussian standard deviation (typically 1.6 times the standard deviation). If the pixel belongs to one of the most probable Gaussians, the pixel is classified as background and the relevant Gaussian parameters (i.e. mean, variance, frequency) are updated. Otherwise, the pixel is classified as foreground and the parameters of the associated Gaussian are updated according to this additional luminance value. At the beginning of the process, a new Gaussian is initialized each time a pixel is classified as foreground until the pre-defined maximum number of Gaussians is reached. The maximum number of Gaussians is a parameter that should

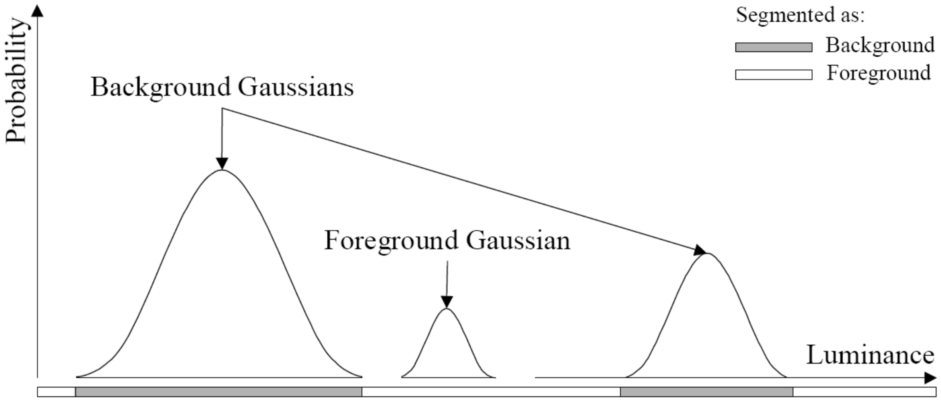


Figure 3.4: *Statistical background modeling of a pixel using three Gaussians. Multiple Gaussians aggregate the pixel luminance values observed in a sliding window.*

theoretically be adapted to the number of different states a pixel of the background could have according to the different noises (acquisition, vibrations, etc.). In practice three Gaussians per mixture perform well in most indoor and outdoor conditions. In order to avoid the construction of Gaussians with flat shapes, pixels are not considered as belonging to a Gaussian if its update would result in a standard deviation greater than ten. This modification of the standard algorithm results in a more reliable modeling of the background.

For each portion of the sliding window, i.e. at any time, an estimate of the background can thus be constructed. It only requires getting the mean of the most probable Gaussian for each pixel. In order to get rid of transient background effects, e.g. when objects stay still and get integrated in the background, the background estimate is updated only in regions where the mixtures of Gaussians are stable. For each pixel, we compute the ratio between the number of occurrences of the most probable and the next probable Gaussians during the last one-second period. We assume that the background modeling of a given pixel can be refreshed when those ratios have not varied by more than 15% for the considered pixel and its neighbours. In our experiments this spatial criteria ensures spatially coherent background estimates.

An example of background frame generated with the proposed algorithm

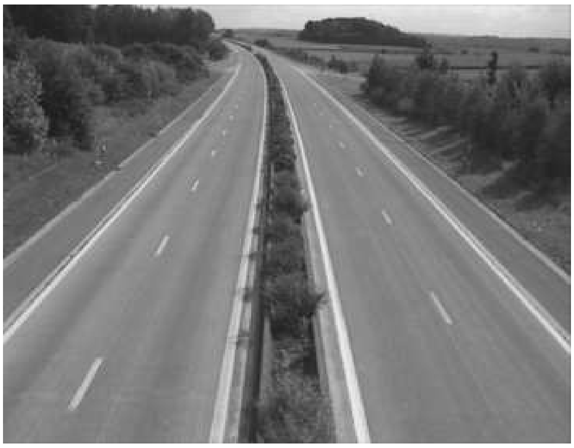


Figure 3.5: *Frame generated with the background estimation process applied to the Speedway sequence.*

is illustrated in Figure 3.5 for the *Speedway* sequence[1].

### 3.5.2 Transmission of background estimates for replenishment purposes

In our replenishment framework, we are interested in providing an alternative reference to the client. Hence, we do not need to transmit the whole sequence of background estimates. Rather, as the background remains constant for a large number of consecutive frames, the background reference is seldom refreshed. In practice, it is updated either at a fixed low frame-rate or only when major background changes are detected.

At the very beginning of the sequence, the background estimate is unstable since the number of aggregated values defining the Gaussians is very small. In order to avoid prohibitive transmissions associated to numerous background updates during this period, the first frame is considered as being the best background estimate until the Gaussian mixtures are considered as stable. In our simulations based on several types of video-surveillance, the background stability is obtained within less than two seconds of video. During this initialization period, a huge part of the scene can sometimes be considered as foreground if many mobile objects are present at the beginning of the sequence or if the sequence is very noisy. While

[1] The *Speedway* sequence is presented in Section 3.7 page 53.

this could be considered as an inherent problem from a strict semantical point of view, it does not have much impact on the delivered video quality within the proposed replenishment method since our approach is based on two reference images.

### 3.5.3 RoI definition

As explained previously in Section 3.4, the rate allocation process can integrate a priori semantic information about the content. Practically, this is achieved by multiplying during the allocation process the precinct distortion values $d(i)$ by a weight $w(i)$ defined for each precinct $i$ by this a priori information (Equation 3.2 page 40). Here, the precinct weights are used to prioritize Regions of Interest (RoI) in the sequence.

In a video surveillance context, Regions of Interest are generally defined to be mobile objects. In some applications, one might be interested only in mobile objects matching pre-defined decision characteristics (e.g. size, position, texture, etc.) or behaviors (e.g. people entering restricted areas).

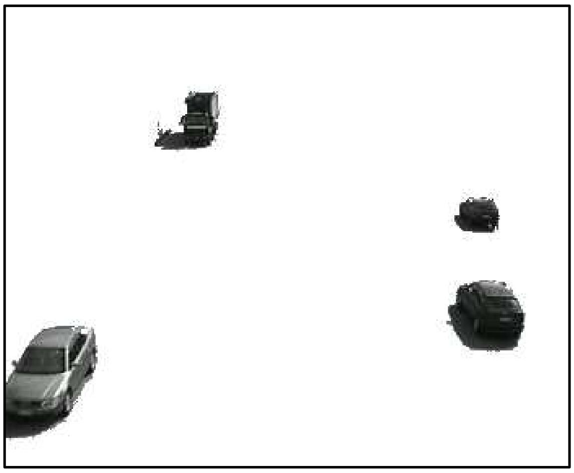


Figure 3.6: *Example of Regions of Interest extracted from the Speedway sequence, at frame 200.*

In our simulations, as in [32], we consider that all pixels classified as foreground by the background estimation algorithm belong to the RoI. One characteristic of the segmentation algorithm is that the background Gaussians widths are automatically adapted to the sequence noise, i.e. the Gaussians have a higher standard deviation in noisy sequences than in sequences with a lower noise. This feature prevents the pixels of a noisy background from being considered as semantically important, and guarantees that the RoI replenishment prioritization will allocate transmission resources to the objects moving in the scene and not to non-relevant variations of background caused by the noise (see Section 3.7.3). Figure 3.6 illustrates the regions that will be prioritized when transmitting a frame of the *Speedway* sequence.

## 3.6 System architecture

The architecture of the proposed conditional replenishment system is depicted in Figure 3.7.

The first operation applied on the source video is the discrete wavelet transform of the frame to transmit. A delay module reconstructs the reference frame. The *Squared Error* (SE) between this reference and the frame to transmit is then calculated. The next step is the generation of parity and JPEG 2000 packets for each precinct, with their associated SE information. Several quality versions of the precincts are encoded, each version characterized by a particular quantization level of the precinct coefficients. These quantization levels are calculated by the optimal JPEG 2000 rate allocation [15] for JPEG 2000 packets and the same levels are used for the parity packets. The way parity bits are generated is explained in Chapter 4. At the top of the figure, the background reference path is depicted. This additional reference is an alternative to the reference provided by the previously decoded frame. This background reference is updated by the background estimation process at regular time intervals or when the background significantly differs from the available background, and is transmitted to the client when updated. A by-product of the background estimation is the Region of Interest (RoI) information, as explained in the previous section. The module selecting the RD optimal replenishment options receives the rate-distortion pairs for each reference, parity and JPEG 2000 replenishment options. Based on this information and the client constraints (resolution, bandwidth, etc.) and preferences (RoI, etc.), replenishment decisions are taken in a RD optimal way as described in Section 3.4. The JPEG 2000 and parity segments are then transmitted to the client, with the replenishment decisions taken for each precinct. After the decoding of these segments, the frame is reconstructed and will serve as a reference for the next frame.

Interestingly, most of these relatively complex operations can be per-

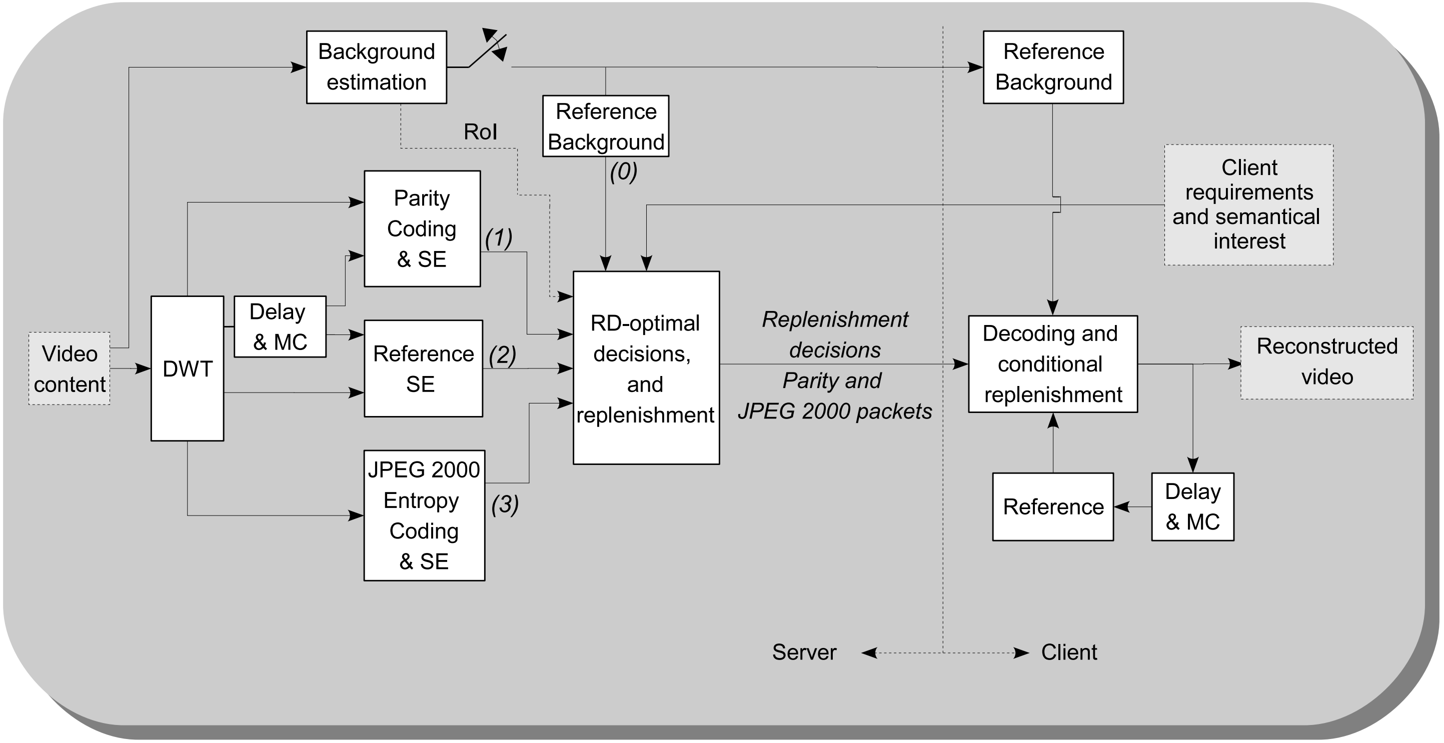


Figure 3.7: ***Overview of the proposed conditional replenishment framework**. A detailed description of the modules and data paths are provided in the text.*

formed only once and offline by the server. This is further discussed and exploited in Chapter 5.

For completeness, an optional *motion compensation* (MC) module is included on Figure 3.7 to reconstruct the reference frame. The goal of the motion compensation module is to improve the reference accuracy. A discussion regarding the design of this module is proposed in the next chapter, at Section 4.6.4. It opens important fundamental questions that have not been investigated in this thesis. Hence, the proposed experiments focus on video-surveillance content with a still background and leaves the study of a motion compensation engine dedicated to our replenishment framework for future research.

## 3.7 Experimental validation

In this section, the conditional replenishment framework is validated in different contexts, and various versions of the system are compared.

We first analyze the performances of the proposed replenishment method when serving a single pre-encoded content at multiple rates. For comparison purposes, we first provide the compression performance achieved by JPEG 2000 and MPEG-4 AVC at similar bitrates. Secondly, we show how the background reference improves the system performances at low rates. We then illustrate how the rate-allocation process can be adapted to favor semantically relevant areas of the content. In the meantime, we demonstrate that the combination of filtered background estimation and RoI-based distortion metric is able to improve the transmitted and reconstructed version of a content initially subject to noise during acquisition. Finally, as a last experimental validation, we analyze the visual quality temporal evolution of different replenishment mechanisms.

Two sequences have been used to generate these results. The first one is *Speedway*, a video-surveillance sequence in CIF format captured from a bridge above a highway, corresponding to a period of time when vehicles are passing in the field of view. *Speedway* has been captured with a fixed camera at 25 fps during 8 seconds and is available on the WCAM european project website [2]. The second one is *Caviar*, a video-surveillance sequence presenting people walking in front of a shop. Its frame-rate, resolution and length is similar to that of *Speedway*. It is available on the CAVIAR project website [1].

Regarding the JPEG 2000 compression parameters, both sequences have been encoded with four quality layers (corresponding to compression ratios of 2.7, 13.5, 37 and 76) and six resolutions. Precinct sizes have been set to 128x128 and code-blocks have a size of 64x64. In simulations where a background reference is used, the compressed background (∼50 kbytes) is sent only once at the beginning of the transmission because it remains sufficiently constant during the sequence duration.

### 3.7.1 Compression efficiency validation

Here, we discuss the compression efficiency of the system by comparing three variants of the proposed replenishment mechanism to conventional JPEG 2000[1] and MPEG4-AVC solutions. The performances of the proposed system and of the JPEG 2000 solution refer to the transmission of a single pre-encoded content at multiple rates. For all replenishment methods, a single reference, provided by the previous decoded frames, is considered. Hence, box (0) is omitted in Figure 3.7.

Figures 3.8 and 3.9 compare the performance of the following coding methods:

- **JR** refers to JPEG 2000 replenishment, which means that boxes (0) and (1) are omitted in Figure 3.7.
- **PR** refers to Parity replenishment, which means that box (3) is omitted in Figure 3.7, in addition to box (0).
- **JPR** refers to JPEG 2000 and Parity replenishment.
- **J2K** refer to the transmission of intra JPEG 2000 frames, which means that box (0), (1) and (2) are omitted.
- **AVC** refers to the transmission of MPEG4-AVC streams with two different Intra Periods (IP). The first method (IP=1) transmits only intra frames, offering a high temporal scalability. The second method (IP=15) encodes an intra frame followed by 14 inter frames, reducing the temporal scalability.

[1] The JPEG 2000 committee has created a video format file encapsulation JPEG 2000 codestreams called Motion JPEG 2000 [27]. Although such format would be used in real applications to limit the number of files to handle and take advantage of the many MPEG compatible metadata boxes, we restrain ourselves in this result section to individual JPEG 2000 codestreams as this does not impact the compression efficiency compared to Motion JPEG 2000.

Regarding the rate control, the bit-budget has been uniformly distributed on all frames for JPEG 2000 and replenishment methods. With respect to AVC, we have adapted the quantization parameters to reach the same average bitrate as for other methods.

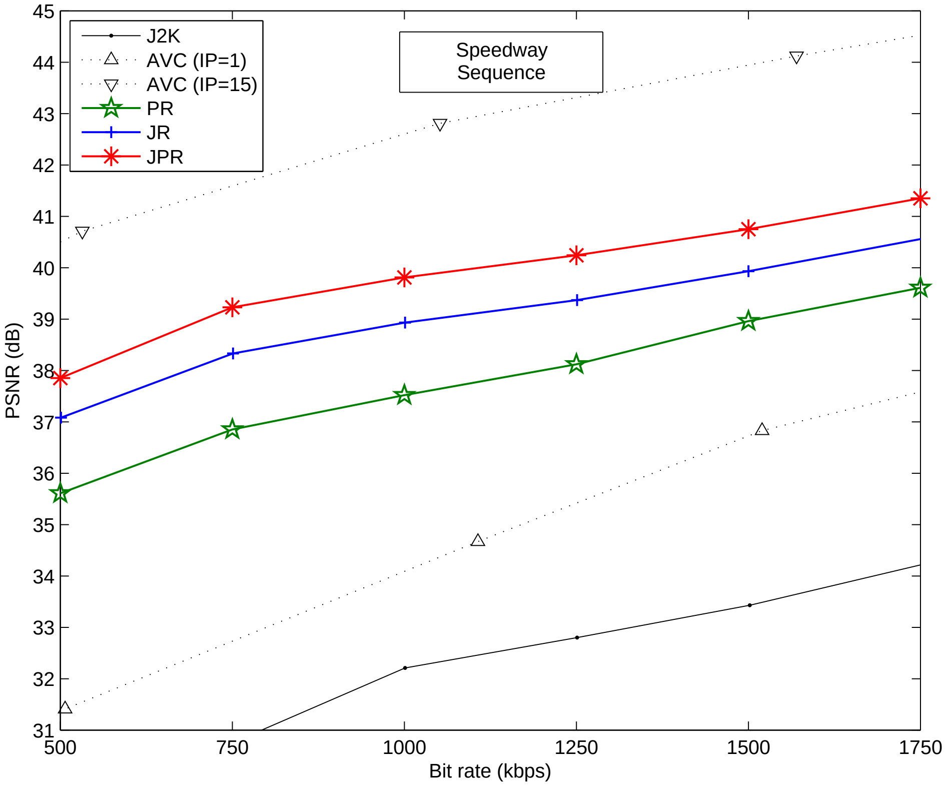


Figure 3.8: *Performances of the proposed system with different combination of Parity and JPEG 2000 replenishments (JR, PR and JPR), MPEG4-(AVC) and the purely INTRA JPEG 2000 coding scheme (J2K) for the Speedway sequence. Frame rates and encoding parameters are defined in the text.*

We observe in Figure 3.8 for the *Speedway* sequence that, unsurprisingly, the standard JPEG 2000 algorithm appears to be the worst scheme from a compression efficiency point of view. J2K is 6-7 dB below PR, which is followed by JR which performs 1 to 1.5 dB better. Finally, the combination of parity bit and JPEG 2000 replenishments improve JR by about 0.8 dB. Compared to MPEG-4 AVC, the replenishment results are convincing, given the increased flexibility offered by these methods and their efficient integration in a low complexity server (see Chapter 5). At 500 kbps, JPR is 6.5 dB above AVC IP-1, and 3 dB below AVC IP-15.

Figure 3.9 presents the same methods for the *Caviar* sequence. We also observe that JPR performs better than JR by about 0.8 dB. However, in this case, PR is not below but above JR, and just below JPR. Hence, JPEG 2000 and parity replenishment are still complementary since their combination exceeds both individual performance, but parity refreshments surpass JPEG 2000 refreshments. This can be explained by the fact that the *Caviar* sequence being less noisy than *Speedway*, the temporal correlation is higher, favoring parity coding as we will see in Chapter 4.

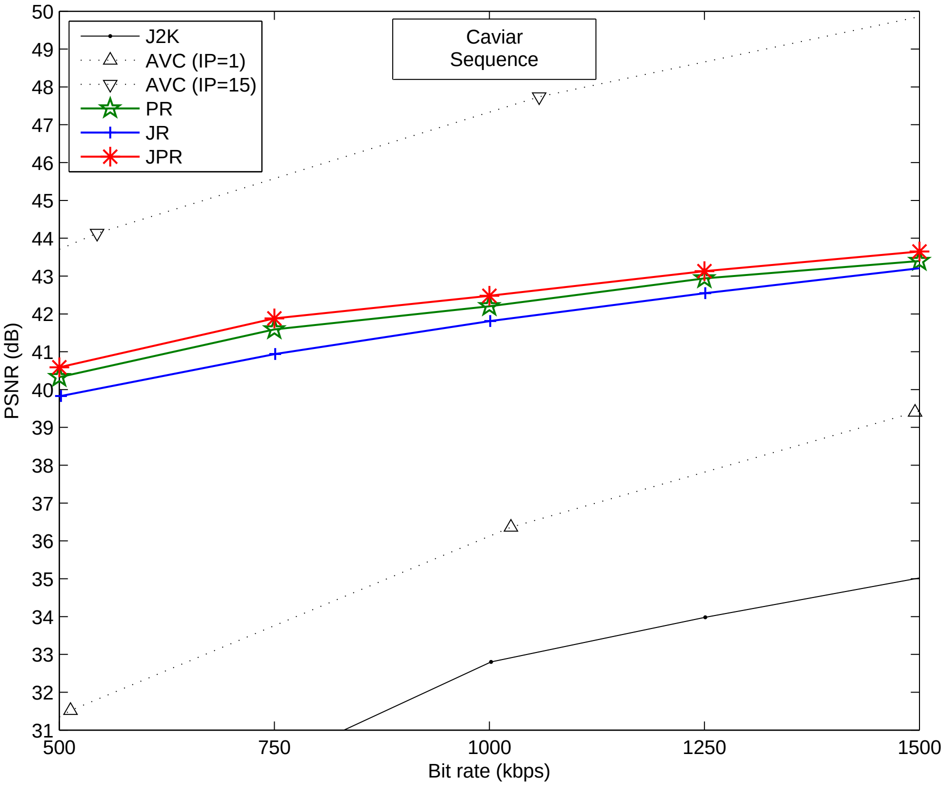


Figure 3.9: *Performances of the proposed system with different combination of Parity and JPEG 2000 replenishments (JR, PR and JPR), MPEG4-(AVC) and the purely INTRA JPEG 2000 coding scheme (J2K) for the Caviar sequence.*

In both Figures 3.8 and 3.9, we observe that the replenishment curves (JR, PR and JPR) are flatter than the AVC and J2K curves. This relative slower increase of quality with the bitrate does not reflect any suboptimality regarding the way the replenishment methods use the available bit budget. Rather, it corresponds to the fact that precincts that get the opportunity to be transmitted at increased rate only moderately improve the quality, compared to the invested bit budget. This is because those precincts were initially approximated based on the reference, at zero transmission cost. As a consequence, the gain in quality is computed with respect to the reference approximation, while the transmission cost is compared to zero. In contrast, for J2K and AVC schemes, an increment of quality induces an increment of bit budget that corresponds to the refinement of already partially transmitted coefficients (finer quantization with AVC or additional layer with J2K), and not to the complete transmission of the information needed to switch from a reference-based approximation to an actual transmission of JPEG 2000 coefficients.

Results involving parity refreshments will be discussed more deeply in the next chapter, after a complete presentation of the parity mechanisms. However, the good results for both sequences of the PR method, solely based on parity replenishments, should be underlined. In the following, we concentrate on replenishment solutions in which parity bits are disabled (box(1) is omitted in Figure 3.7).

### 3.7.2 Analysis of the benefit provided by the background reference

In this section, we analyze the benefit obtained when considering a second reference candidate, defined to be an estimate of the scene background as described in Section 3.5.1 (corresponding to box (0) in Figure 3.7).

In addition to JR, J2K and AVC, we thus consider the following coding method:

- **JRB** refers to JPEG 2000 conditional replenishment with background. Hence, only box (1) is omitted in Figure 3.7. This method proposes to consider both the previous image and the estimated background as possible references for each precinct. In practice, for a given precinct, the reference that best approximates the precinct is selected for that specific precinct.

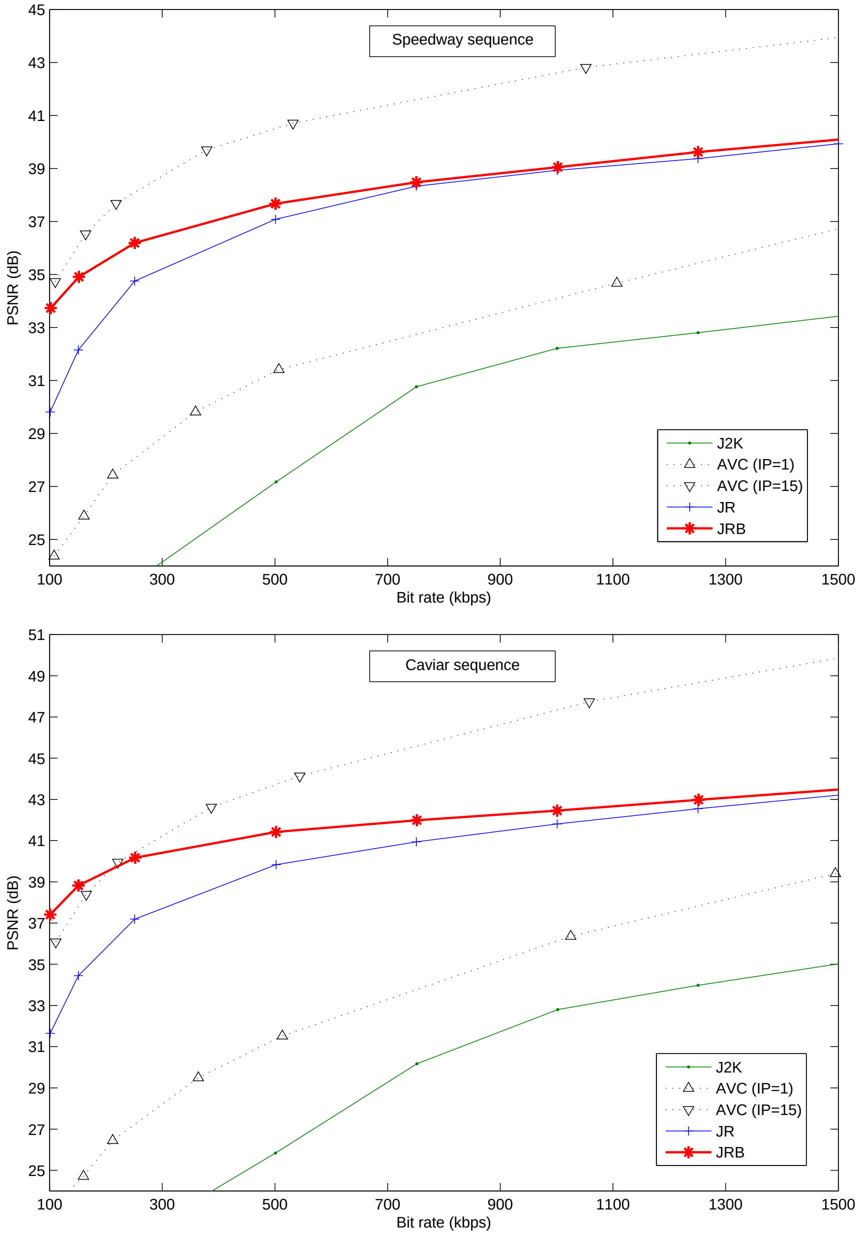
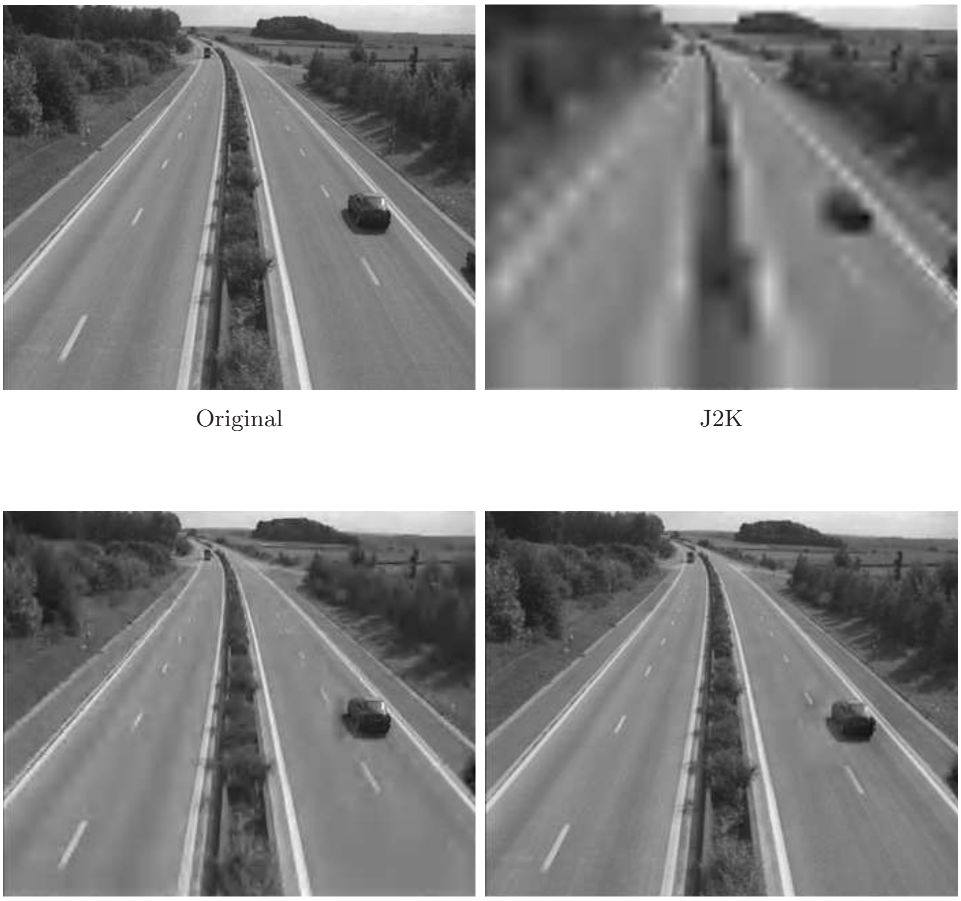


Figure 3.10: *Rate distortion curves of the JR, JRB, J2K and AVC methods for the Speedway (upper graph) and Caviar (lower graph) sequence.*

Figure 3.10 presents the rate distortion curves for the *Speedway* and *CAVIAR* sequences. We will focus on the *Caviar* sequence for the following analysis. The methods represented on this figure are JR, JRB, J2K and AVC.

At very low bitrates (100 kbps), the JRB method improves J2K by 20 dB and JR by more than 5 dB. The difference between JR and JRB tends to decrease with the bitrate, as the relative gain brought by the background approximation decreases.



Original

J2K

JR

JRB

Figure 3.11: *J2K, JR, and JRB methods for the 10th frame of the Speedway sequence transmitted at 250 kbps.*

In order to figure out the meaning from a perceptual point of view of the RD curves, Figure 3.11 presents snapshots of the *Speedway* sequence compressed with the J2K, JR, and JRB methods at 250 kbps. We observe that at this low bitrate, JR considerably improves the J2K method, and still remains blurry compared to JRB.

### 3.7.3 Semantically weight adaptive streaming

We now consider two scenarios for which the server adapts its packet scheduling decisions to the specific interest expressed by the client about the scene content. In both scenarios, moving objects are considered to be more important than the scene background. In the first scenario, this knowledge is used to prioritize the replenishment of moving objects. In the second scenario, the same knowledge is exploited to mitigate the impact of a noisy content acquisition process on the replenishment decisions. Both scenarios illustrate the flexibility of the proposed replenishment method, and its ability to integrate individual user needs at transmission time, based on a single pre-encoded codestream.

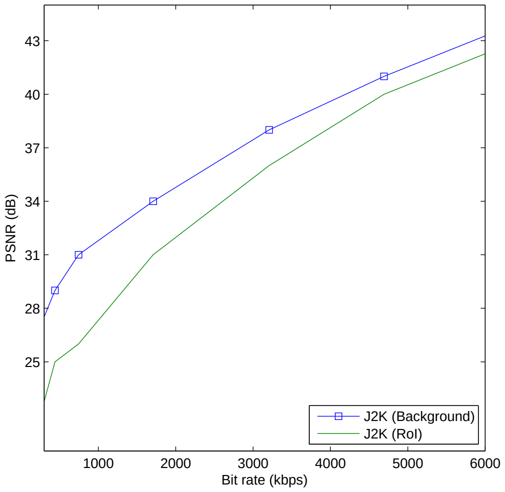
We introduce the following coding method:

- **JROI** refers to JPEG 2000 conditional replenishment with Regions of Interest (dashed arrow in Figure 3.7). This method follows the mechanism introduced by JRB, but defines the distortion based on a weighted SE of wavelet coefficients (see Section 3.3.3), in order to take into account the knowledge the server might have about the semantic significance of approximation errors. In practice, much knowledge can be provided based on user feedback or on some kind of automatic pre-analysis of the scene. Here we assume that the information about the semantic relevance of approximation errors is provided at the precinct level based on a foreground object extraction.

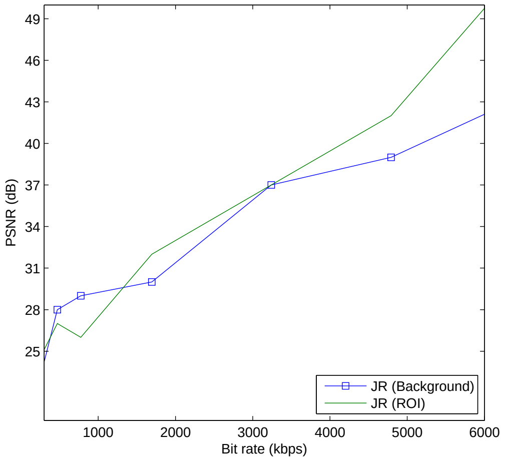
To maximize the impact of RoI prioritization, the semantic weights $w(i)$ defined in Section 3.3.3 are set to one (zero) for precincts that belong to the RoI (background) areas. The strategy is aggressive but defines a limit case that enables us to get a clear idea about the potential benefit to draw from a semantic weighting of distortion.

Note that for these results, we consider that a precinct belongs to the RoI if at least 5% of its supporting pixels are labeled as foreground RoI pixels. The supporting pixels of a precinct are obtained by dyadic upsampling of the precinct subband support.

### RoI-based streaming

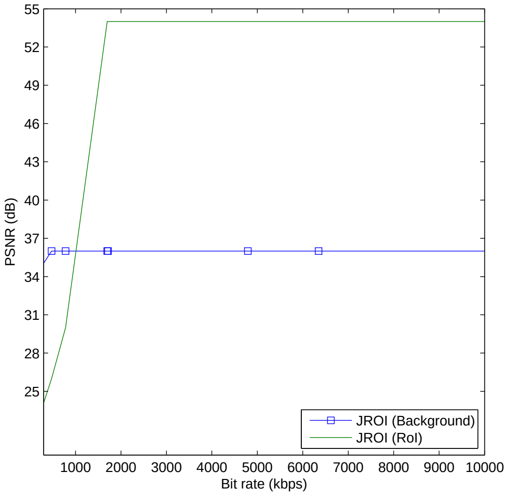


J2K method

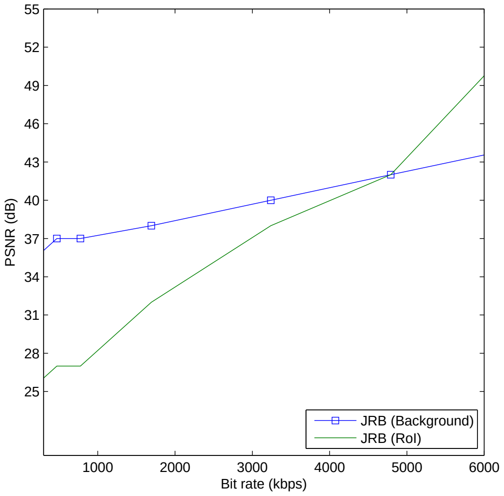


JR method

Figure 3.12: *RoI and background quality as a function of the total transmission rate for the J2K and JR methods (Speedway sequence).*



JROI method



JRB method

Figure 3.13: *RoI and background quality as a function of the total transmission rate for the JROI and JRB methods (Speedway sequence).*

Figure 3.12 and 3.13 present the PSNR of RoI and background regions of *Speedway* for several conditional replenishment mechanisms. We observe that, for the J2K method, the background quality is always higher than the RoI because most of these background regions, such as the road and the sky, are very efficiently compressed. Indeed, since these regions are quite predictable, the JPEG 2000 entropy coder easily reduces the number of bits used to code them compared to regions with a lower predictability. The RoI contains the cars that are characterized by a large amount of details, which are less efficiently compressed.

Compared to J2K, the JR method offers a higher quality for the RoIs, which correspond to the zones that are more often replenished. This trend is reinforced in the JROI method, which rapidly maximizes the RoI quality but maintains constant background quality. This is explained by the fact that the background areas are never replenished by JROI, and are only defined using the background reference transmitted at the same high quality for all bitrates. Note that in our simulation, once the RoI reaches its maximal quality, JROI does not transmit additional data to improve the background region, even if some bit-budget is available. The JRB method behaves like JR at high bit rates, but offers a higher background quality at low bit rates, since the background reference can be used to increase background quality.

### Sequence with acquisition noise

In this paragraph, we consider a noisy version of the *Speedway* sequence to further illustrate the flexibility of our proposed streaming server. Specifically, we show that our proposed method naturally supports the exploitation of a priori knowledge about the relevance of approximation errors in the scene.

In the scenario considered here, we have added white Gaussian noise with a standard deviation of 10 to the *Speedway* sequence[1], as illustrated in Figure 3.14. The noise simulates the effect of adverse surveillance conditions: noisy camera acquisition, bad weather, presence of traffic lights or moving objects (trees, ...). Note that we do not consider here noise due to transmission errors, but noise originated in the acquisition.

[1] The values of the sequence components are integers between 0 and 255, and the corresponding SNR is 23.4 dB.

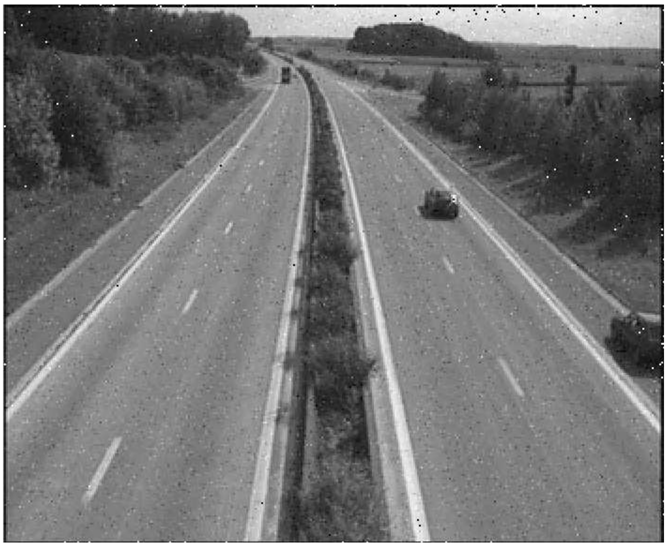
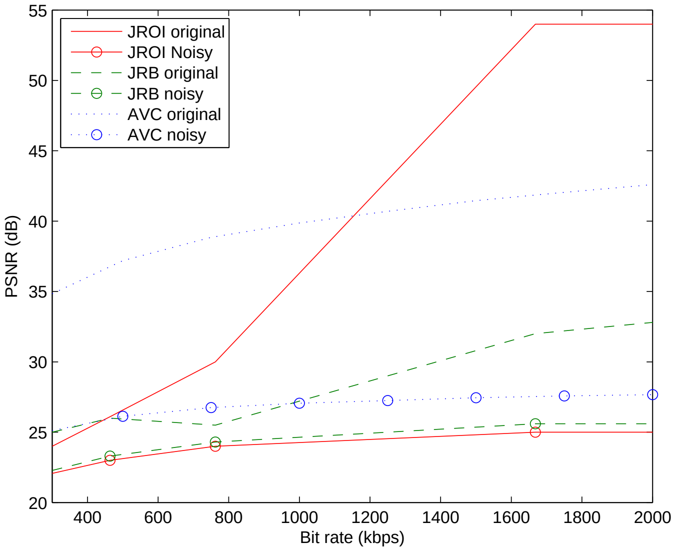


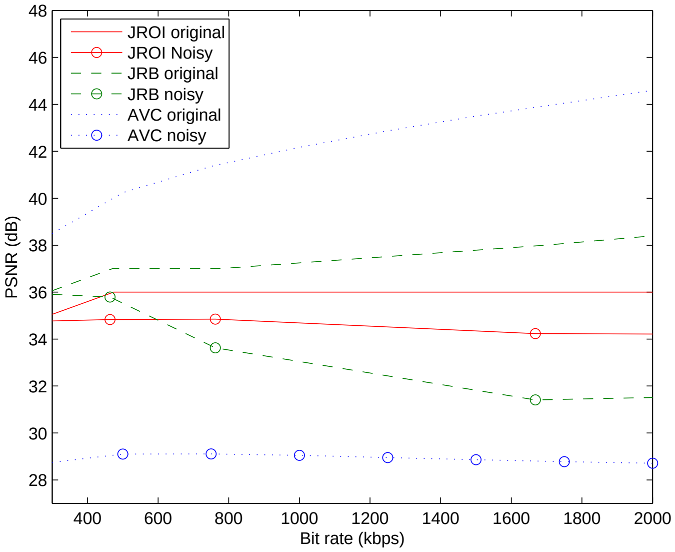
Figure 3.14: *Snapshot of the Speedway sequence corrupted with additive white Gaussian noise characterized by a standard deviation of 10.*

The noise causes luminance changes in the background regions, but these changes are not relevant with respect to the surveillance purpose of the application and should not trigger replenishment mechanisms. Hence, the approximation error observed on background areas should be neglected compared to errors measured in the foreground moving areas. In our simulation, this is simply done by using the JROI method, with distinct weights assigned to foreground and background precincts. Indeed, one characteristic of the segmentation algorithm presented in Section 3.5.1 is that the background Gaussians widths are automatically adapted to the sequence noise, i.e. the Gaussians have a higher standard deviation in noisy sequences than sequences with a lower noise. This feature prevents the pixels of the background to be considered as foreground pixels, even in case of strong noise, which in turns guarantees that the RoI replenishment prioritization allocates transmission resources to the objects moving in the scene, and not to the non-relevant variations of background caused by the noise.

Moreover, the background estimation process filters the sequence temporally and provides a denoised version of the background. Thus, we expect the JROI method to offer a denoised, and perceptually more pleasant version of the sequence at the client side. This is confirmed visually and illustrated in Figure 3.15 where the original sequence is taken as a reference



RoI



Background

Figure 3.15: *RoI and background quality for the JROI, JRB and AVC methods in normal and noisy conditions (Speedway sequence). The PSNR is calculated using the original (non noisy) sequence as reference.*

to compute the PSNR values obtained when transmitting the original and noisy sequences based on the JROI, JRB and AVC methods, respectively.

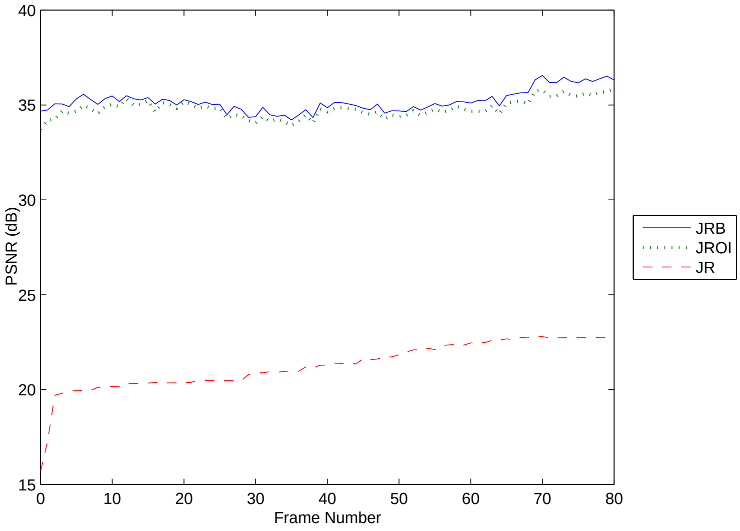
The upper part of Figure 3.15 focuses on the *RoI*. When dealing with original (non noisy) content, all transmitted bits of the JROI method are dedicated to the RoI, which explains the higher performances of this method compared to JRB, and even to AVC for sufficiently large rates. In noisy conditions, the RoI quality of all methods sharply decreases since it is computed with respect to the original sequence, while all codecs attempt to describe the noise.

The lower part of Figure 3.15 represents the *background* quality. In normal conditions, AVC outperforms JRB and JROI. In more details, the JROI method minimizes the rate allocated to background regions, thereby preventing the background quality to increase with the bit budget. In contrast, JRB progressively refreshes the background regions as the global (RoI + background) available rate increases, providing a higher overall background quality.

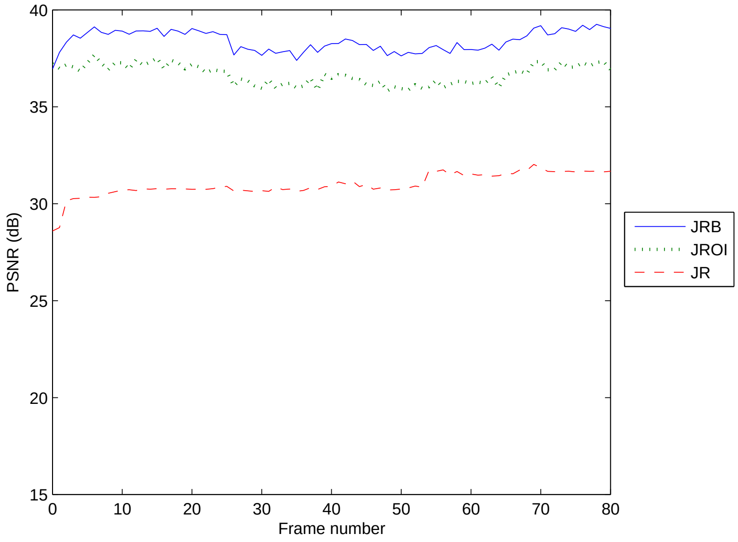
In noisy conditions, we observe that JROI outperforms both JRB and AVC. Since the background regions are modified by the noise at each frame, the JRB (AVC) method regularly refreshes (corrects prediction errors for) those regions, mainly to render noise effects, which ends up in decreasing the quality compared to the original signal. On the contrary, since the JROI method knows a priori that most of the changes affecting the background are due to noise, it concentrates the refresh on RoI regions and almost never refreshes the background regions, thereby providing a higher background quality compared to the original (without noise) sequence. The same argument also explains why the background quality -measured with respect to the original sequence- is higher than the RoI quality when considering the JROI encoding scheme. In short, RoI noise is coded accurately while a denoised filtered background is used as the reference for the background, resulting in a background signal which is closer to the original.

### 3.7.4 Temporal evolution of the quality

For completeness, figure 3.16 shows the temporal evolution of the quality for the JR, JROI and JRB methods. We observe that the quality offered by these methods is quite constant during the transmission. At low bit rates, the JR quality slightly increases until frame 70. This is due to the fact that, at those bit rates, the background blocks are slowly replenished



*Speedway* at 235 kbps



*Speedway* at 1600 kbps

Figure 3.16: *Temporal evolution of the image quality for the JRB, JROI and JR methods (Speedway sequence transmitted at 235 kbps and 1600 kbps).*

compared to the other methods. The JRB approach introduces a peak of bit-consumption at the beginning of the session due to the transmission of the estimated background. The JROI method behaves similarly to JRB.

### 3.7.5 Error resilience capabilities

In this section, we propose a preliminary analysis of the replenishment system performances in a noisy environment and propose simple methods to validate our intuition. We do not consider conditional replenishment with parity bits but restrict ourselves to replenishment with JPEG 2000 packets. Parity bits are expected to bring an increased robustness to the system since these bits are not only able to correct virtual channel errors, but also to correct errors created by the transmission channel [22]. However, this is out of the scope of this work but is certainly an interesting perspective for future research.

The conditional replenishment transmission framework is characterized by the fact that the refreshed information is transmitted in INTRA, without any reference to the past. Hence, it naturally provides some resilience to transmission errors, since an error only remains perceptible until the next successful refresh. Unfortunately, this assertion also means that areas that are rarely refreshed become more sensitive to transmission errors than other regions. In order to prevent persistent errors when transmitting video in noisy environments, a particular attention should thus be devoted to those regions that become temporally stable after a period during which they were significantly changing. Indeed, for those regions, if the last refreshment before a stable period is lost, then the resulting reconstruction error affects the whole stable period, with dramatic perceptual impact.

The problem formalization should be done in a rate distortion framework, similarly to what we have done in [8] for the resilient transmission of JPEG 2000 codestreams, and incorporate temporal considerations related to the variability of the temporal impact of errors in our replenishment framework. The optimization of the replenishment scheduling taking into account this variability is beyond the scope of this paper. However, we provide an illustrative example based on an adaptive scheduling and a heuristic protection which retransmits important refresh.

To validate our intuition, Figure 3.17 considers an error-prone channel characterized by independent and identically distributed (iid) bit errors, and assumes that a packet is lost as soon as one of its bits becomes erroneous. The figure compares three scheduling methods. The first one is the conventional JPEG 2000 replenishment method (*JR*). The second one, denoted *JR Robust I*, knows the channel BER and takes it into account to schedule JPEG 2000 packets. Specifically, it uses a first order approximation to compute the reference distortion, and accounts for the packet loss probability to compute the benefit expected from JPEG 2000 packets transmissions. The third one, denoted *JR Robust II*, extends the previous method by adding a simple heuristic to improve the robustness of critical refresh.

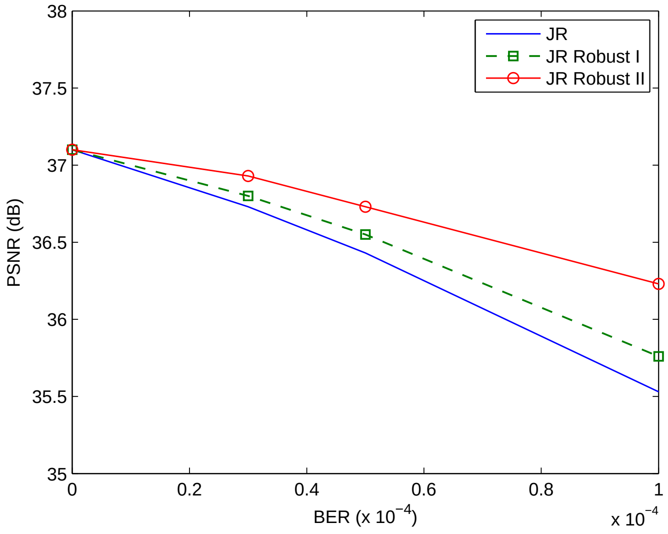


Figure 3.17: *Comparison of three replenishment methods as a function of the channel bit error rate for the Speedway sequence at 500 kbps. Transmitted JPEG 2000 packet are considered as lost as soon as one of their bits becomes erroneous. The three methods are described in the text.*

Formally, let us denote $(k_1, q_1)$ and $(k_2, q_2)$ the time indexes and quality levels associated to the two latest refreshment of the $i^{th}$ precinct. Let us also denote $d_t^{k_1,q_1}(i)$ and $d_t^{k_2,q_2}(i)$ the distortion measured when approximating the $i^{th}$ precinct at time $t$ either based on the last or last but one refreshment, and $d_t^{ref}(i)$ the distortion measured based on the reference available at time $t$. If we denote $p_1$ the probability that the last refreshment of precinct $i$ at time $t$ has been lost, the first order approximation of the reference distortion can be computed as $d_t^{ref}(i) \cong (1 - p_1)\, d_t^{k_1,q_1}(i) + p_1\, d_t^{k_2,q_2}(i)$. It corresponds to a first order approximation since it ignores the fact that the last but one refreshment might also be lost. Similarly, the benefit expected from the refreshment of the $i^{th}$ precinct at time $t$ can be estimated based on the knowledge of the channel BER. Those refinements of the expected distortions are implemented by the *JR Robust I* method.

However, they are unable to prevent the appearance of persistent errors in regions that become stable, e.g. after a moving object discloses a still background. This is due to the fact that $p_1$ is typically small, making $p_1\ d_t^{k_2,q_2}(i)$ insignificant compared to changing areas in the frame. To circumvent this drawback, we propose a simple heuristic to identify the regions that are expected to remain stable for some time after significant changes, and force an additional refreshment for them, so as to ensure with a high probability that they will be correctly received at the client. In practice, this is done by setting $d_t^{ref}(i)$ to $d_t^{k_2,q_2}(i)$ for the regions for which $d_t^{k_2,q_2}(i) \gg d_t^{k_1,q_1}(i)$, which are regions that appear to be significantly changing before $k_2$ and stable at time $k_1$. The curve *JR Robust II* in Figure 3.17 implements that heuristic. Unsurprisingly, we observe that it significantly improves the resilience of the conditional framework to losses. We conclude after this preliminary analysis that an adaptation of the scheduling algorithm should enable the replenishment framework to efficiently support error-prone channels.

## 3.8 Conclusion

In this chapter, we have presented the proposed framework combining four replenishment options. The two first options are references that can be used to approximate the segment to transmit. The first reference consists in the previous decoded frame and the second in an estimate of the frame background. When these references offer a poor approximation of the precinct to transmit, it can be refreshed by a JPEG 2000 or a parity packet.

These replenishment decisions are taken in an optimal rate-distortion framework, based on distortion metrics that can integrate at transmission time information regarding the transmission conditions and user preferences in the content browsed content, like regions of interest.

The remarkable feature of our system lies in the fact that the adaptation of forwarded content to user needs and resources is performed without requiring to generate and manipulate multiple encoded versions of the same content, and as we will see in Chapter 5, at low computational cost.

The proposed coding model brings interesting advantages when compared to closed-loops frameworks. Although a reference is also exploited in our work, the data transmitted are INTRA JPEG 2000 packets or parity bits which tolerate a certain desynchronization between the encoder and the decoder. This is particularly important for transmissions in error-prone environments, as well as when serving heterogeneous clients with different prediction references.

# Parity based replenishment 4

*The content of this chapter is basically the reproduction of [18].*

*Our replenishment system presented in Chapter 3 offers two coding methods to replenish a precinct that is not correctly estimated by the reference. These two methods are JPEG 2000 and parity coding. When the first method is selected, the reference is refreshed by decoding the corresponding JPEG 2000 packet. In the second case, the reference is corrected with parity bits. This chapter describes the way our system generates and allocates these parity bits to image precincts, and exploits temporal and spatial correlation in the source.*

## 4.1 Introduction

The reference precinct, typically extracted from the previous decoded frame, constitutes the main component of the side information that is exploited to encode the current precinct using parity bits. In this chapter, we analyze how the temporal correlation between the side information and the precinct to encode can be formalized in order to improve the correcting efficiency of the parity bits. Similarly, we discuss how the nature of the data that is transmitted - images - can be exploited during the decoding stage.

This chapter is structured as follows. We first present how the principles of video coding using side information have been adapted to our conditional replenishment framework. Then, we explain how to exploit the temporal correlation between consecutive frames of wavelet bit-planes and the spatial correlation inherent to an image source. We then detail the practical implementation of the parity replenishment module. Finally, the system performances are presented and discussed.

## 4.2 Conditional replenishment with side information

We have presented in Section 2.6 the principles of video coding with side information. We will see in this section how we have adapted these principles to the conditional replenishment framework presented in the previous chapter.

Figure 4.1, which has already been presented in Section 2.6, depicts video coding systems with side information. Parity bits are generated at the encoder based on $X$, the frame to transmit. With these parity bits, the decoder corrects the side information $Y$, which is usually the previous decoded frame, and generates the reconstructed frame.

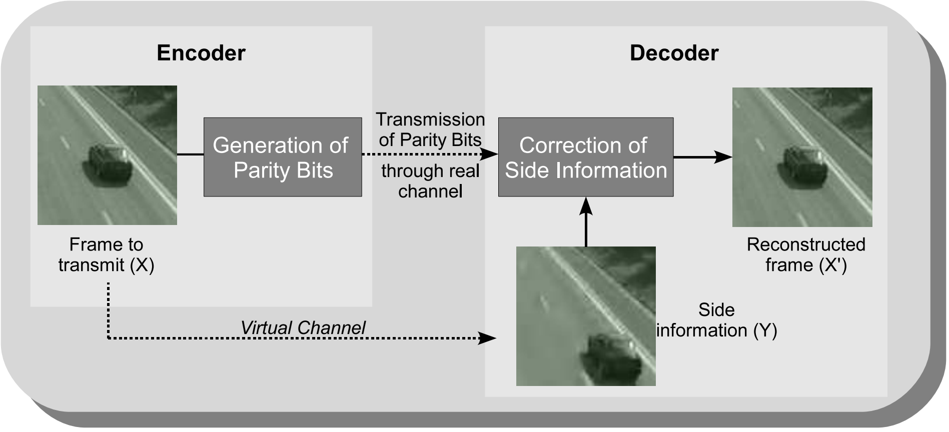


Figure 4.1: *Video coding system with side information.*

We have seen in Section 2.6 that the rate at which we can transmit $X$ depends on two factors:

- *The mutual information between $X$ and $Y$.* This value will be high if $X$ can be efficiently predicted from $Y$. This can be done by exploiting the temporal correlation between the reference and the image to transmit, and will be studied in Section 4.3.
- *The entropy of $X$.* In practice, the frame $X$ is encoded based on codewords that are shorter than the frame size. Encoding those codewords independently most often result in a significant increase of entropy,

compared to $H(X)$. Hence, it is important to exploit the correlation between the codewords of $X$ so as to maintain the entropy of the actual codeword source close to the initial frame entropy $H(X)$. In our work, this is achieved by representing $X$ through spatially localized subband samples and by exploiting the frequency and spatial correlation between those samples, as studied in Section 4.4.

In our system, parity bit replenishments must preserve the granular access to the compressed data in terms of spatial access, resolution and quality, since a major motivation of this work is scalability. Hence, parity bits are generated at the encoder independently for each precinct, ensuring spatial and resolution scalability. The precincts are encoded in several embedded quality layers, each layer gathering the parity bits correcting a certain number of consecutive bit-planes, thereby preserving the scalability in quality.

To describe the system formally, we follow conventional notations, and denote random variables with upper cases. Their realization is denoted with corresponding lower cases. Bold fonts are used to denote a vector of random variables.

Of all practical error correction methods know to date, LDPC [23] and Turbo codes [10] come closest to approaching the Shannon limit. In this work, although the same could be achieved with other channel codes, we focus on LDPC codes. LDPC codes are characterized by a transformation matrix $\mathbf{H}$ of size $M$x$K$. At the encoder side, the sequence of input bits belonging to the precinct to transmit is considered as a random vector $\mathbf{X}$ of length $K$ and is mapped into its corresponding $\mathbf{Z}$ parity bits of length $M$, achieving a compression ratio of $K : M$.

At the decoder side, let $K$ be the length of the random vector $\mathbf{Y}$ corresponding to the bits of the reference precinct, which are combined with the received parity bits $\mathbf{Z}$. The initial probability distribution of the $i^{th}$ reference bit $Y_i$ can be defined in different ways depending on the way the temporal correlation is modeled. This is described in Section 4.3.

The main goal of the decoder is to exploit the source model and the parity bits $\mathbf{Z}$ in order to iteratively converge toward input bits $\mathbf{X}$.

Our decoding model is illustrated in Figure 4.2 as a *factor graph* [38]. A factor graph is a bipartite graph that expresses which variables (circles) are arguments of which local functions (squares). The local functions $f_i$

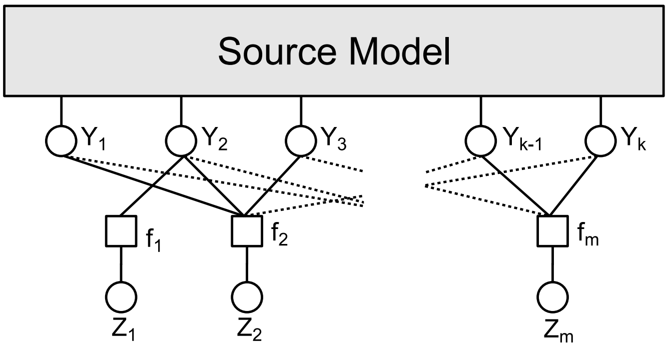


Figure 4.2: *Factor graph representing the source model and the LDPC code. At each iteration of the decoding process, messages describing local distributions are exchanged between nodes inside the LDPC code and with the source model.*

represent the linear transformation **H** and the results of that transformation are the parity bits variables $Z_i$. Our graph consists here of two main components: the source model and the LDPC code. The source model is detailed in Section 4.4. It takes into account the spatial correlation between precinct coefficients during the decoding process.

Decoding is achieved using the sum-product algorithm [38]. This algorithm aims at computing the marginal posterior probabilities $P(Y_i = 1|\mathbf{Z}, \mathbf{H})$ for each $i$. This is achieved by iteratively transmitting between nodes messages corresponding to an estimate of the local variable distribution and updating each node with the received information.

## 4.3 Exploiting temporal correlation in the wavelet domain

This section describes how temporal correlation between successive frames is exploited to initialize the probability distribution associated to $Y_i$ variables. We first define a model for the temporal correlation between the wavelet coefficients of consecutive frames, and then explain how to translate this correlation between coefficients into probability distributions for the $Y_i$ variables, which by definition correspond to the bits of wavelet coefficients.

### 4.3.1 Gaussian distribution of coefficients

For simplicity, we adopt a simple model to describe the temporal correlation between corresponding coefficients of two consecutive frames[1]. Typical models used to describe such correlation follow the Laplacian or Gaussian distributions. In this work, we have adopted the latter one.

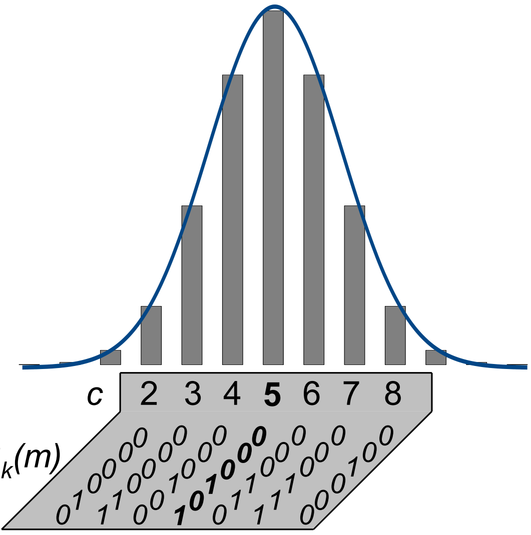


Figure 4.3: *Probability distributions of coefficients have to be adapted due to the representation in bit-planes. Bits representing the coefficient are weighted assuming a Gaussian distribution centered on the coefficient.*

Formally, let $C^{n,r,t}$ denote the random variable associated to the $n^{th}$ wavelet coefficient of the $r^{th}$ resolution at time $t$. We assume that the corresponding coefficient at time $t+1$ follows a Gaussian distribution of variance $\sigma^2_{r,t}$ around the realization of $C^{n,r,t}$, and the probability distribution of $C^{n,r,t+1}$ is defined by:

$$P\left(C^{n,r,t+1} = m | C^{n,r,t} = n\right) = \frac{1}{\sqrt{2\pi\ \sigma^2_{r,t}}}\ e^{-\frac{(n-m)^2}{2\ \sigma^2_{r,t}}} \tag{4.1}$$

Based on the coefficient distribution, we can compute the distribution

[1] Note that the correspondence between coefficients might account for a potential motion vector, when a motion field is defined between consecutive frames. This is not the case in our system which omits motion compensation (see Section 4.6.4)

for $B_k^{n,r,t+1}$, the random variable associated to the $k^{th}$ most significant bit of $C^{n,r,t+1}$. For this purpose, we introduce $\beta_k(m)$ to denote the function extracting the value of the $k^{th}$ bit of coefficient $m$. Hence, we have $B_k^{n,r,t+1} = \beta_k(C^{n,r,t+1})$ and

$$P(B_k^{n,r,t+1} = 1 | C^{n,r,t} = n) = \frac{1}{\sqrt{2\pi\ \sigma_{r,t}^2}} \sum_{m=-\infty}^{\infty} e^{-\frac{(n-m)^2}{2\ \sigma_{r,t}^2}} \ \beta_k(m) \qquad (4.2)$$

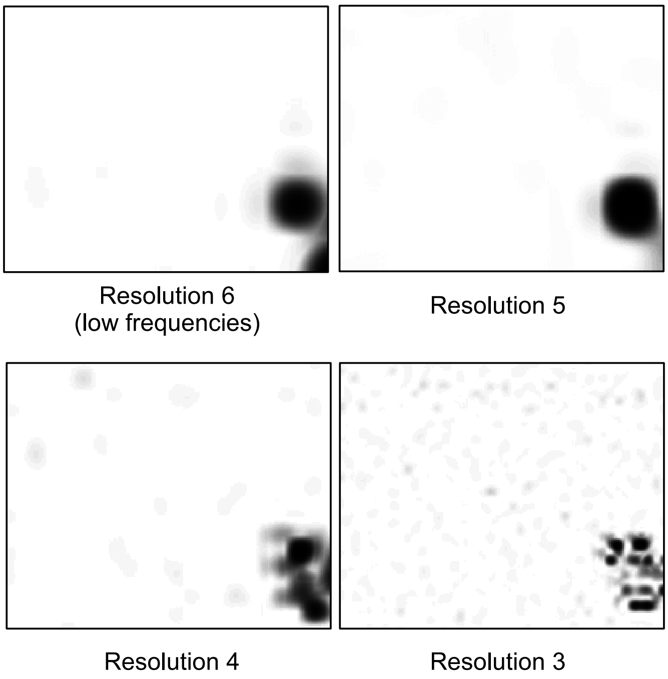
Figure 4.3 illustrates this Gaussian weighting of the coefficients bits. In the chosen example, non-significant MSB have a very low probability of becoming significant, while the uncertainty on the value of the LSB is very high.

### 4.3.2 Variance estimation

The variance parameter is estimated for each resolution by the coefficients mean squared prediction error in this resolution. For simplicity, we consider in the following that the variables are taken at time $t$ ($C^{n,r,t}$ is now denoted $C^{n,r}$) and denote the variance parameter for resolution $r$ by $\sigma_r^2$. This value is computed for each frame at the encoder and transmitted to the decoder.

This approximation can be spatially refined by taking into account the spatial variations of this variance. Indeed, important modifications of the content between consecutive frames are reflected in the concerned spatial regions throughout several resolutions. This is illustrated in Figure 4.4 which presents spatial maps of absolute prediction errors for distinct resolutions of the *Speedway* sequence. We observe that the wavelet coefficient differences between consecutive frames are spatially relatively coherent through resolutions. This spatial coherence decreases as frequencies increase, because of the presence of noise, and due to the fact that smaller content modifications mostly impact high resolutions.

Hence, a coefficient will have more chance to change from one frame to another if the coefficients belonging to the same spatial zone in the lower frequency resolution have changed. This observation can be integrated in the proposed system by defining the variance as a function of the coefficient



Resolution 6 (low frequencies)

Resolution 5

Resolution 4

Resolution 3

Figure 4.4: *Maps of absolute difference between resolution coefficients in two consecutive frames, by decreasing order of resolutions starting from the low frequency resolution on the left, for first two frames of the Speedway sequence. In this figure, resolutions have been resampled to the same size and the absolute difference values of each resolution have been rescaled between 0 (white regions) and 255 (black regions).*

index, based on the evolution of corresponding coefficients in the lower resolution.

At the decoder, precincts are decoded in increasing order of frequency (hence starting with precincts belonging to the low frequency resolution). In this way, the decoding of coefficient $C^{n,r}$ can benefit from spatial information from the neighborhood of the coefficient $C^{n,r+1}$, in the lower frequency resolution. We introduce $L_{\sigma}^{n,r+1}$ which evaluates the local variance of coefficients in the neighborhood of $C^{n,r+1}$ and is calculated as a weighted sum of $E_{sq}^{n,r+1}$, the coefficients squared prediction error, the weight being proportional to the neighbor distance to the coefficient $C^{n,r+1}$.

If we denote $d_{m,n,r}$ the absolute distance between coefficient $C^{m,r}$ and coefficient $C^{n,r}$, the local variance $L_{\sigma}^{n,r}$ is defined as:

$$L_{\sigma}^{n,r} = (\sum_{m=-\infty}^{\infty} \frac{1}{d_{m,n,r}})^{-1} * \sum_{m=-\infty}^{\infty} \frac{E_{sq}^{m,r}}{d_{m,n,r}} \tag{4.3}$$

Formally, the variance of coefficient $C^{n,r}$ is defined by

$$\sigma_{n,r}^{2} = \sigma_{r}^{2} \frac{L_{\sigma}^{n,r+1}}{\sigma_{r+1}^{2}} \tag{4.4}$$

which means that the variance of coefficient $C^{n,r}$ is evaluated by the variance of its resolution weighted by the relative modifications observed on the neighborhood of the corresponding coefficient in the lower frequency resolution.

The benefits of the local refinement of the variance estimation is discussed in Section 4.6.2.

## 4.4 Exploiting spatial correlation of precincts

This section explains how the spatial correlation of precinct coefficients can improve the correction of the reference. First we present the model of our source. We then detail how this source model can be integrated in the sum-product algorithm.

### 4.4.1 Source model

The source model aims at capturing the statistical behavior of a source, which is an image in our case. It mainly exploits the fact that the value of a coefficient inside a precinct is highly correlated to its neighbors.

In our work, the *image modeler* is based on the Embedded Block Coding with Optimized Truncation (EBCOT) algorithm [67], used in the JPEG 2000 standard [3] and already presented in Section 2.3.2. We recall here the important notions related to the EBCOT that are useful for this chapter.

According to this algorithm, a bit is classified in one of the 19 different categories called *contexts*, based on the significance[1] of its eight contiguous

[1] A bit is considered as significant if at least one bit belonging to a higher bit-plane in the same coefficient has its value set to 1.

neighbors and its own significance state. The statistics of the bits can differ highly from one context to another. The way contexts are calculated depends on its subband since this has an influence on the way bits are spatially correlated.

Bits labeled with the same context have the same neighborhood and hence are characterized by a similar statistical behavior. This similar statistical behavior is exploited during the decoding by providing a soft estimation of each bit based on its neighborhood.

### 4.4.2 Source model integration in the sum-product algorithm

In practice, our system takes advantage of the image modeler as follows. The decoder exploits the context bit probability distributions by integrating this soft information in the sum-product algorithm. We will first see how these distributions can be calculated, and the explain the integration in the sum-product algorithm.

Formally, we denote by $\mathscr{C}(i)$ the context computed by the EBCOT algorithm around the $i^{th}$ bit. The probability distribution of bit $Y_i$, knowing its context, is the written $P(Y_i = 1|\mathscr{C}(i) = c)$, where $c$ is the context index of the $i^{th}$ bit.

The bit probability distributions defined for each context can be calculated in different ways. First, this distribution can be estimated based on histogram computations, i.e. frequencies of occurrences at the encoder, and transmitted to the decoder. This implies a significant transmission overhead but offers the best context statistics for the precinct to transmit. At the opposite, statistics pre-calculated on a large set of precincts belonging to different types of images can be hard-coded at the decoder. This solution avoids transmissions and computation, but offers generic context statistics instead of precise statistics based on the particular precinct to decode. Another alternative consists in computing at the decoder side these statistics based on the corresponding precinct in the previous frame. To decide between these three alternatives, we now analyze the temporal evolution of the context statistics.

Figure 4.5 presents the temporal evolution of the probability distribution of the nine first contexts, for the lowest resolution of the five first frames of the *Speedway* sequence. We observe that context distributions remain quite constant for most contexts. This observation remains valid

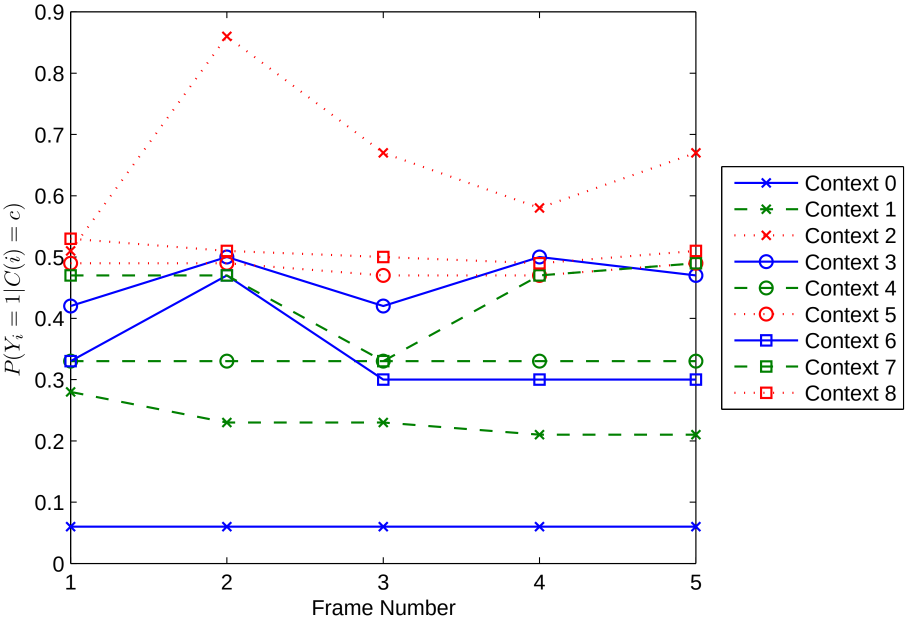


Figure 4.5: *Temporal evolution of $P(Y_i = 1|\mathscr{C}(i) = c)$ for the nine first contexts for the lowest resolution. These statistics have been calculated on the five first frames of the Speedway sequence.*

for higher resolutions. At the light of this graphic, an alternative to the solutions presented above consists for the encoder to restrict the transmission to distributions that have significantly evolved since the previous frame. In our simulations, we have however decided to rely on a hard-coded distribution, computed based on a representative set of images. We now explain how those distributions are used by the parity-bits decoder.

At each iteration of the sum-product decoding algorithm, a hard decision is taken for each variable $Y_i$ and passed to the source modeler. The modeler calculates the context index number $c = \mathscr{C}(i)$ corresponding to the hard decisions taken for the neighbors of $Y_i$ and returns the soft information $P(Y_i = 1|\mathscr{C}(i) = c)$, as illustrated in Figure 4.6. The probability of $Y_i$ is then updated for the next iteration of the sum-product algorithm. If soft information was provided to the EBCOT about $Y_i$ instead of hard information, the modeler could alternatively calculate a weighted sum of probabilities associated to all possible contexts.

Regarding complexity, we consider as a first approximation that the integration of the source model in the sum-product algorithm roughly doubles the number of operations carried out during the decoding process. Indeed, we consider that the complexity related to the EBCOT context computation for each bit is approximately equivalent to the complexity of the bit probabilities update performed during the sum-product algorithm.

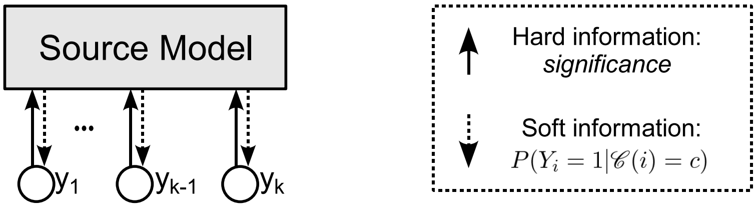


Figure 4.6: *Hard decisions are transmitted from each LDPC node and soft informations calculated in the Source Modeler are sent back.*

**SUMMARY OF EXPLOITED CORRELATION**

| | |
|---|---|
| **Temporal Correlation** | ⇒ *Gaussian distribution of coefficients* |
| **Coherence across resolutions** | ⇒ *Gaussian variance estimation* |
| **Spatial Correlation** | ⇒ *EBCOT context modeling* |

## 4.5 Practical implementation

This section gives practical details regarding the way the parity replenishment module has been implemented.

### 4.5.1 Raw coding of bit-planes

The correlation between bit-planes of corresponding precincts in successive frames usually decreases with the bit-plane significance. When the correlation is under a certain threshold, it is more efficient to transmit the bits in a raw mode than to try to reconstruct the bit-planes with parity information. In the following results, the threshold value has been set to a bit error rate between corresponding bit-planes in consecutive images equal to 0.15. This threshold value is heuristic and could obviously be refined based on a sharper analysis of parity bits efficiency combined with the spatial correlation information.

### 4.5.2 LDPC codes

In our system, a limited number of LDPC matrices **H** have been generated. For each precinct to encode, the smallest of these matrices able to correct the reference precinct has been selected, and the generated parity bits transmitted. In Chapter 5, we describe how this choice can be adapted to individual clients and low computational cost.

We have considered regular LDPC codes with a standard bipartite graph structure [42]: the columns weight have been set to three, and the weight per row as uniform as possible. The cycles of length four in the factor graph representation of the code have been eliminated [35].

A maximum value of 8000 bits has been defined for the codeword length $N$. This value represents a compromise to limit the system complexity while offering efficient LDPC codes [13, 58]. Hence, for large precincts, several codewords characterized by the same codeword length $N$ and parity length $M$ are generated and interleaved as described in the next paragraph.

The generation and storage of these multiple **H** matrices could be avoided by using fountain codes [43], like the LT [40] and Raptor codes [62], in place of LDPC codes.

### 4.5.3 Interleaving

Since the parity codeword length is limited, several codewords must be generated for large precincts. In this case, the last message which contains the parity bits correcting the LSB usually has a higher BER[1] than the other messages. Since a single matrix **H** is selected for each quality layer of a precinct, the number of parity bits $M$ is constant for all the messages encoded to a precinct layer. For this reason, interleaving of the precinct bits **X** between the several messages is implemented to make the BER of all messages uniform. This is illustrated in Figure 4.7.

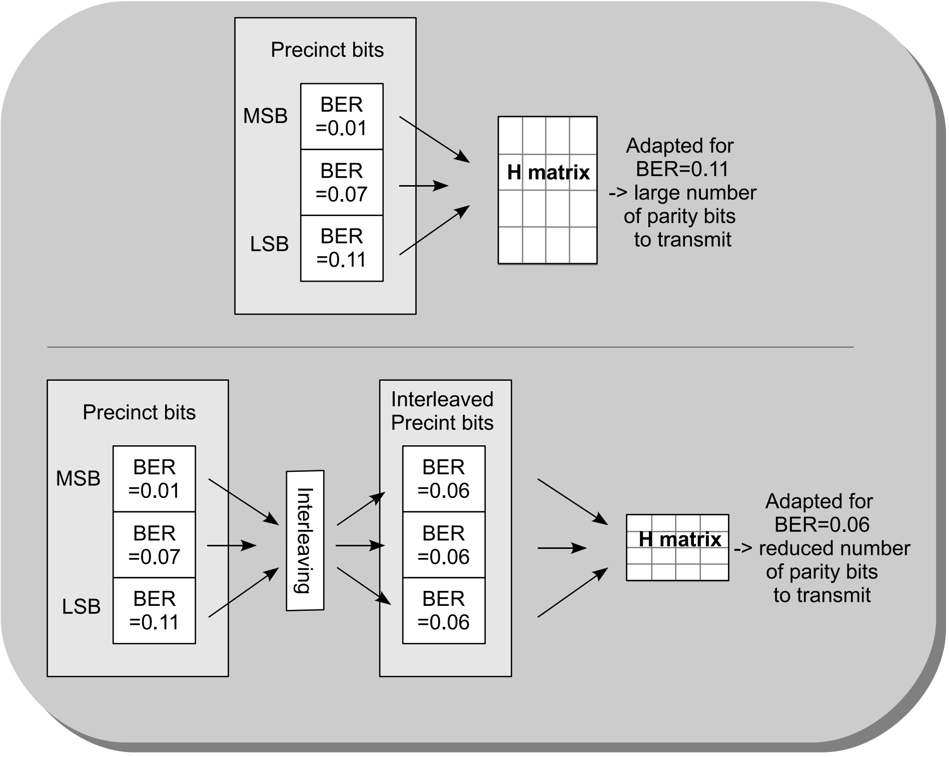


Figure 4.7: *When multiple codewords are generated for a given precinct, interleaving of the precinct bits enables to reduce the size of the matrix H required to correct the precinct, reducing the number of parity bits to transmit.*

[1] The Bit Error Ratio (BER) considered here is related to the virtual channel between $X$ and $Y$, that we assume binary and symmetric.

### 4.5.4 Contexts

Practically, only a subset of the 19 contexts defined by the EBCOT [3] are considered in the simulations. Our system considers the nine contexts for the significance propagation and cleanup coding passes, the five sign contexts, as well as the three contexts for the magnitude refinement coding passes. Run-lengths and uniform contexts are not considered, as they are not relevant to our system.

## 4.6 Results

In this section, we first recall the global performances of the replenishment system integrating parity refreshments. Then, we present the performances reached when exploiting the temporal and spatial correlations. At the light of the spatial correlations results, we finally discuss a number of aspects that should drive the future design of a motion compensation module.

### 4.6.1 Global results

In this section, we analyze the performances of the proposed system when transmitting a single content at multiple rates.

These results have been generated with the *Speedway* and *Caviar* sequences. Their characteristics have been presented in Section 3.7 page 53. Figure 4.8 and Figure 4.9 compare the performance of our system for both sequences. The coding mechanisms considered are *JPEG 2000* Replenishment (JR), *Parity* Replenishment (PR) and *JPEG 2000 and Parity* Replenishment (JPR) as defined in Section 3.7.1. These results have already been partly discussed in Section 3.7, but we focus here on the parity-bit refresh mechanism.

We observe as expected that JPR, which combines both refresh mechanisms, offers the best performance for both sequences. This confirms the fact that JPEG 2000 and parity replenishments are complementary. Implemented alone, the JPEG 2000 and parity replenishments behave differently in these sequences. In *Speedway*, the JR performs better than PR and the opposite happens for the *Caviar* sequence. This can be explained by the fact that *Speedway* is characterized by a relatively low but constant acquisition noise. In this context, the correlation between frames is lower than in

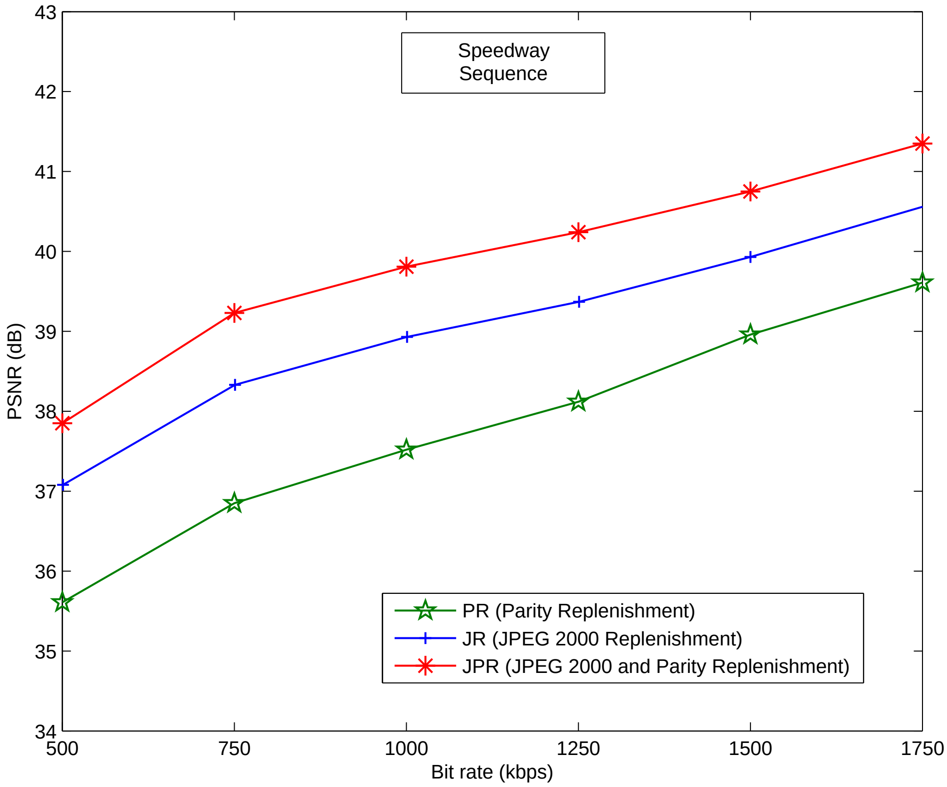


Figure 4.8: *Performances of the proposed system with different combination of Parity and JPEG 2000 replenishment for the Speedway sequence.*

the *Caviar* sequence which is not noisy. Hence, the efficiency of parity-bits is reduced in the *Speedway* sequence.

When analyzing more deeply the replenishment decisions taken in the JPR method, we observe that parity bits are mostly used at low resolution, where the temporal correlation is high. At higher resolutions, JPEG 2000 replenishments are chosen because of the efficiency of the entropy coding engine, specially when dealing with long runs of zero coefficients.

A deeper analysis of the chosen replenishment options is provided in Table 4.1. This table presents the replenishment options selected for each precinct at various bitrates, when both parity and JPEG 2000 mechanisms are activated. We observe that at low resolutions, the parity mechanism is always more efficient than JPEG 2000. In intermediary resolution (resolutions 2, 3 and 4), both mechanisms are selected, depending on the bitrate.

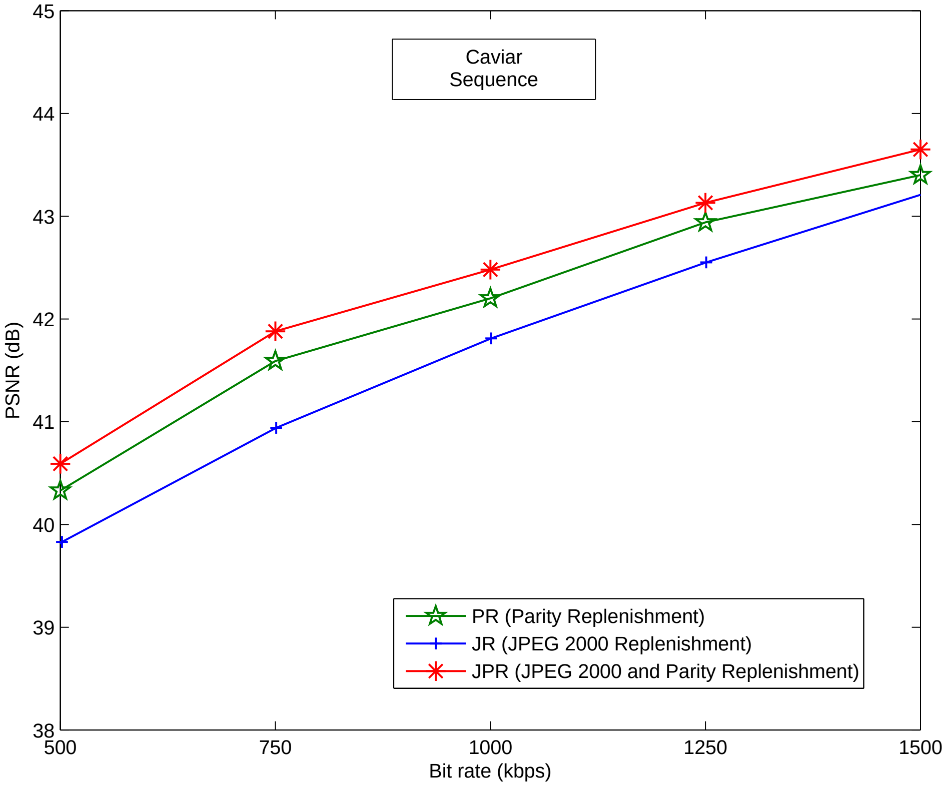
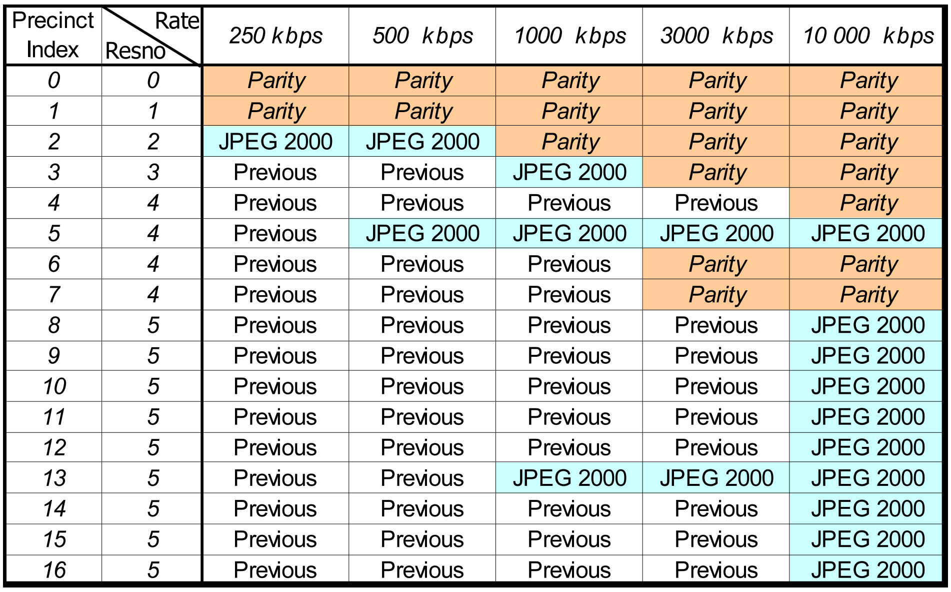


Figure 4.9: *Performances of the proposed system with different combination of Parity and JPEG 2000 replenishment for the Caviar sequence.*

At the highest resolution, JPEG 2000 is the unique replenishment method used. These results demonstrate that the two mechanisms are complementary. At low resolutions, due to the high temporal correlation, parity refresh are more efficient than JPEG 2000. This relative efficiency decreases with the increasing resolutions.

When analyzing more in detail the RD graph of the precinct number 6, which is not represented here, we observe that its convex-hull starts at low bitrates by the reference option, passes by a JPEG 2000 refresh and finally goes through two parity refresh of increasing quality at high bitrates. This is reflected in Table 4.1 by the fact that at low bitrates, the previous precinct is used for the replenishment, followed by parity replenishments. With a finer bitrate granularity in the table, we would have observed the JPEG 2000 refresh between the these two replenishment options (between



| Precinct Index | Rate<br>Resno | 250 kbps | 500 kbps | 1000 kbps | 3000 kbps | 10 000 kbps |
|---|---|---|---|---|---|---|
| 0 | 0 | Parity | Parity | Parity | Parity | Parity |
| 1 | 1 | Parity | Parity | Parity | Parity | Parity |
| 2 | 2 | JPEG 2000 | JPEG 2000 | Parity | Parity | Parity |
| 3 | 3 | Previous | Previous | JPEG 2000 | Parity | Parity |
| 4 | 4 | Previous | Previous | Previous | Previous | Parity |
| 5 | 4 | Previous | JPEG 2000 | JPEG 2000 | JPEG 2000 | JPEG 2000 |
| 6 | 4 | Previous | Previous | Previous | Parity | Parity |
| 7 | 4 | Previous | Previous | Previous | Parity | Parity |
| 8 | 5 | Previous | Previous | Previous | Previous | JPEG 2000 |
| 9 | 5 | Previous | Previous | Previous | Previous | JPEG 2000 |
| 10 | 5 | Previous | Previous | Previous | Previous | JPEG 2000 |
| 11 | 5 | Previous | Previous | Previous | Previous | JPEG 2000 |
| 12 | 5 | Previous | Previous | Previous | Previous | JPEG 2000 |
| 13 | 5 | Previous | Previous | JPEG 2000 | JPEG 2000 | JPEG 2000 |
| 14 | 5 | Previous | Previous | Previous | Previous | JPEG 2000 |
| 15 | 5 | Previous | Previous | Previous | Previous | JPEG 2000 |
| 16 | 5 | Previous | Previous | Previous | Previous | JPEG 2000 |

Table 4.1: *Replenishment mechanisms selected for each precinct at various rates for the first replenished frame of the Speedway sequence.*

1000 kbps and 3000 kbps).

Quite surprisingly, in precinct number 2, the JPEG 2000 refresh are more efficient than the parity refresh at low bitrates. This can be explained by the fact the first quality layers of this precinct have few bit-planes. Hence, the number of bits to transmit is low, and the parity codewords are short. With such short parity codewords, the LDPC compression efficiency is much lower than JPEG 2000. However, at higher bitrates, the transmission of higher quality layers with more bit-planes is possible. As confirmed in the last columns of the table, parity coding is more efficient than JPEG 2000 in this case.

### 4.6.2 Temporal correlation

The way temporal correlation has been exploited in our work has been presented in Section 4.3. First, a Gaussian distribution has been proposed to model the temporal evolution of coefficients. It has then been observed that the coefficients temporal evolution is spatially coherent across resolutions. To integrate this observation into the the Gaussian model, a spatial refinement of the Gaussian distribution variance has been adopted.

Figure 4.10 illustrates the benefits obtained from these two solutions to exploit the temporal correlation. These results have been generated on a portion of the *Speedway* sequence, and only parity replenishments are considered (PR method in Section 3.7.1). The curve labeled "*Standard initialization*" corresponds to the standard initialization usually encountered in video coding systems with side information. In this case, the BER between precincts is measured at the encoder, and is used in combination with the reference precinct bits to initialize the probabilities during the LDPC decoding. The second curve "*Simple Variance*" correspond to the integration of the Gaussian distribution. For third curve labeled "*Predicted Variance*", the system adapts the Gaussian variance based on the prediction error of the corresponding coefficients observed in the lower resolution. Each curve brings an increase of about 0.3 dB compared to the previous at low bitrates and about 0.7 dB at high bitrates. At 1500 kbps, the exploitation of temporal correlation brings a gain of 1.5 dB.

### 4.6.3 Spatial modeling with EBCOT

As presented in Section 4.4, spatial correlation is exploited based on a spatial image modeler, the EBCOT. Various experiments have been realized to determine in which context this modeler improves the system performances.

It rapidly appeared that spatial modeling does not improve the performances when the reference is consistent with the EBCOT model, which is the case when the reference is close to a natural image. In this case, the reference image statistics are very similar to the targeted image statistics. Hence, the refinement of the code-block bits probabilities achieved by the EBCOT is not significant, and no improvement in the compression efficiency is observed.

However, when the reference image available at the client has suffered a degradation, its statistics do not always correspond to an image[1]. In this case, the EBCOT detects these incoherences and helps the LDPC decoder by correcting the bits probabilities that do not respect natural images statistics.

To better understand the role of the EBCOT, we have added increasing random errors to the reference code-block coefficients, focusing either on

[1] This might for example occur when the prediction results from motion compensation.

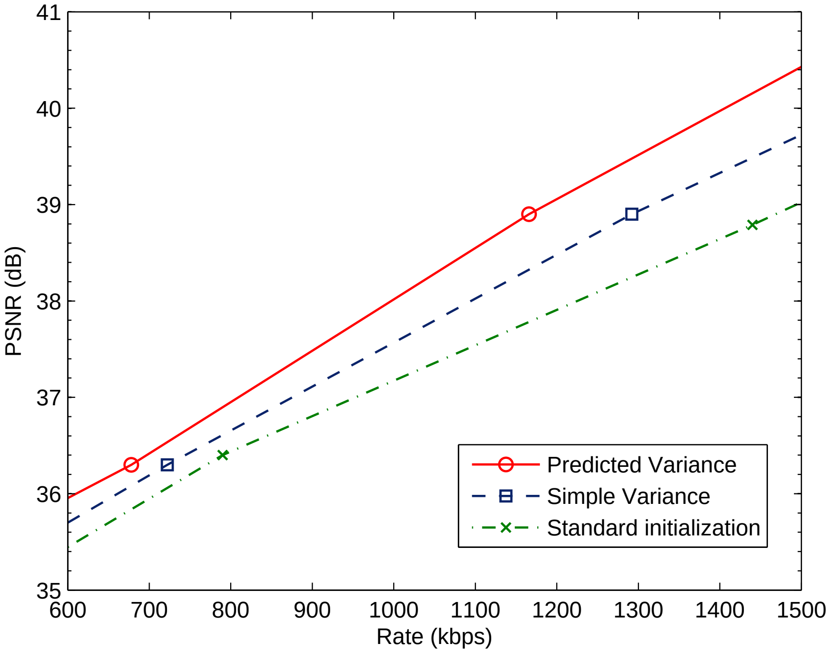


Figure 4.10: *Comparison of the three methods exploiting the temporal correlation between frames in the wavelet domain.*

LSB, or on MSB bit-planes[1], and have analyzed how the length of parity codes increases with error rate, with and without EBCOT. The results of these simulation are presented in Figure 4.11. The figure illustrates the evolution of the gain in compression length offered by the EBCOT when errors are added to the reference LSB and MSB bit-planes, respectively. The abscissa of the graph represents the coefficients error energy, meaning that for a given position on the $X$ axis, the number of errors on the LSB will be much higher than on the MSB. The outcome of this figure is obvious: the EBCOT improves the system performances mainly when the reference contains errors in the MSB.

This is explained by the fact that spatial correlation is high in these bit-planes, while bits belonging to lower bit-planes are less predictable. This is confirmed by the analysis of the refinement contexts which are the contexts

[1] Practically, code-blocks bit-planes have been divided in two groups of equal size: LSB bit-planes and MSB bit-planes.

of coefficients that have become significant in earlier bit-planes. The probabilities associated with these contexts are very close to 0.5, meaning that very few information can be expected from the neighborhood. This is also in line with the observations made when analyzing the special JPEG 2000 coding mode designed to reduce the coding complexity and called the *bypass mode* [3]. With this mode enabled, only the four MSB bit-planes of the code-block are entropy encoded. The remaining bit-planes are raw-coded in the bit-stream. We have shown in [17] that this mode does not impact significantly the compression performances (less than 2% decrease in compression efficiency), confirming the fact that few correlation can be extracted from low bit-planes.

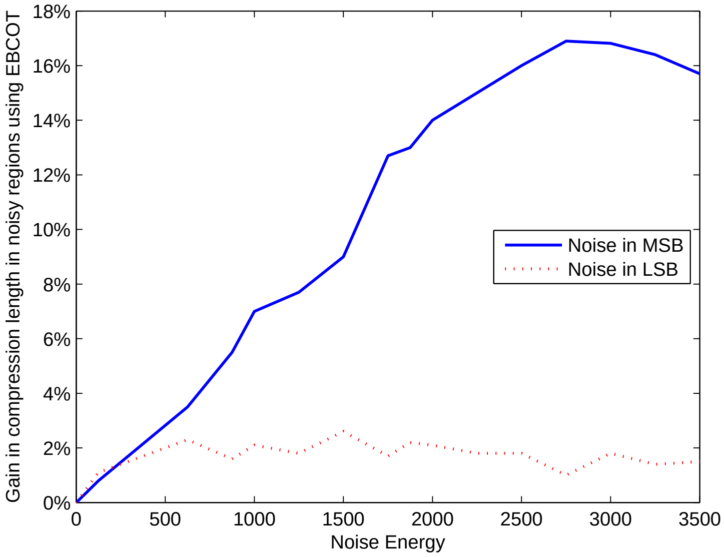


Figure 4.11: *Spatial modeling increases the system performances when the reference available at the client is erroneous. This figure illustrates the increase in compression efficiency brought by the spatial modeler when noise is added to the reference LSB and MSB bit-planes. These results have been generated using code-blocks of the third resolution of the Speedway sequence.*

A deeper analysis of Figure 4.11 reveals that the gain provided by the EBCOT in the MSB tends to saturate and even decrease when error rates increase. This is due to the fact that when an error occurs, the bit contexts are affected. With a high number of errors, contexts of erroneous bits are also erroneous, preventing the EBCOT to offer a correct prediction.

We will see in the next section how metrics used in the motion estimation module can be chosen to create errors which are efficiently corrected by the spatial modeler.

Note that the above result is surprising, as spatial modeling has been used successfully in other works dealing with related coding contexts, and such conclusions have never been drawn. We now explain the main differences between these works and ours. The EBCOT has for example been integrated in a joint source-channel image coding system [22]. The main difference with our system comes from the fact that in that work, which aims at coding individual images, no reference is used to initiate the decoding, and image statistics are thus welcome to help the turbo decoding. In our case, the system is initiated with a coherent reference image, characterized by image statistics related to the image to decode. Hence, the benefit to draw from an image source model is reduced.

A second difference comes from the spatial scalability requirement. In the case of [22], spatial scalability is not required and precincts can be much larger than in our system. With large precincts, another division of the wavelet coefficients into parity packets can be done. In the case of [22], a parity packet aims at correcting entire bit-planes. Hence, before decoding a given bit-plane, all the previous bit-planes are correctly decoded. In that case, the system can entirely trust the context value of a coefficient, which is calculated based on previous bit-planes. In our case, since several bit-planes are decoded simultaneously, the confidence in the context of a coefficient is limited and reduces the benefit to draw from the EBCOT.

Hence, our parity decoder with spatial modeling prefers a small number of errors with a high magnitude than many errors of small magnitude.

### 4.6.4 Preliminary investigation of the design of a motion compensation module in a parity replenishment context

In the previous section, we have explained that the image modeler is mostly efficient when the reference mainly differs from the signal to encode in high-magnitude coefficients. This observation is of primary importance regarding the design of a motion compensation module, since the underlying motion estimation engine has some freedom in shaping the prediction error based on appropriate selection of motion vectors.

Our parity decoder with spatial modeling prefers a small number of errors with a high magnitude than many errors of small magnitude. Hence, the motion estimation algorithms should prioritarily reduce the spatial extent of the errors. In other words, when possible, the motion estimation engine should prefer a prediction that results in few errors of high amplitude rather than a prediction that introduces smaller but more frequent errors. By modifying the metrics used in the motion estimation module, we can favor the prediction option that correspond to localized errors of important magnitude.

Figure 4.12 illustrates two common metrics used in image and video coding: The MSE (Mean Squared Error) used for rate allocation and SAD (Sum of Absolute Differences) used in motion estimation. As we are looking for a metric favoring few errors with a high magnitude, we propose to use another metric, the SRAD (Sum of Rooted Absolute Difference) for this task. Compared to SAD and MSE, the distortion relative to small errors is increased and large errors bring a reduced distortion with the SRAD metric.

The design of a motion compensation that would exploit the specificities of our proposed system has not been considered in this thesis but is certainly an interesting research perspective.

## 4.7 Conclusion

This chapter presents an attempt to complete the conditional replenishment framework with the paradigm of coding with side information. A reference frame, typically the previous frame, constitutes the main component of the side information that is exploited to encode the current frame.

To preserve compatibility with JPEG 2000 intra coding, the side information has to be exploited in the wavelet transform domain. Hence, a particular attention has been devoted to the definition of a practical coding framework that is able to exploit the temporal but also spatial correlation among wavelet subbands coefficients, while defining the parity bits on subsets of bit-planes to preserve quality scalability.

With that respect, three original proposals have been made in this chapter, namely

1. The temporal prediction of individual bits of wavelet coefficients through a Gaussian coefficient distribution formalism,

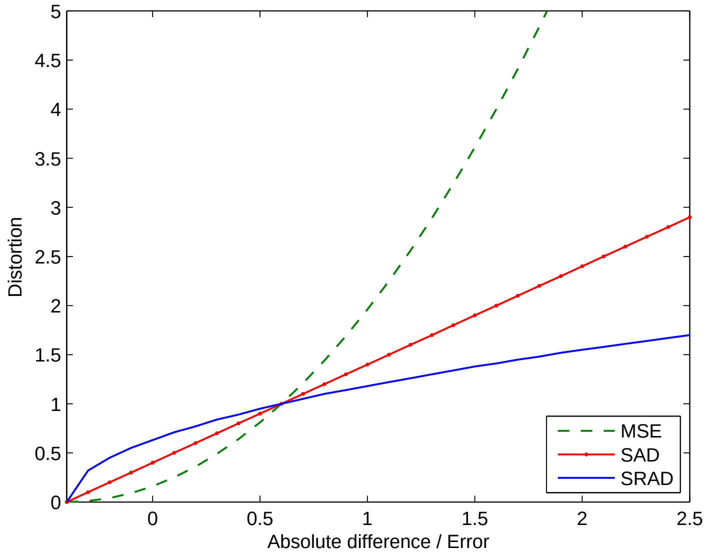


Figure 4.12: *Different metrics for absolute differences. The MSE (Mean Squared Error) used for rate allocation is compared to SAD (Sum of Absolute Differences) usually used in motion compensation, and to the SADR (Sum of Absolute Difference Root). The SADR, which is proposed in this work, is expected to create motion estimation errors that will be efficiently corrected by the spatial modeler.*

2. The spatial adaptation of the Gaussian variance based on the correlation inherent to adjacent resolutions,

3. The exploitation of spatial correlation through context-based bitplane prediction and iterative decoding strategies. The findings related to this exploitation of spatial correlation are important since they should drive the design of the motion-compensation module, which is a mandatory step to extend our replenishment system to any kind of moving content.

These three mechanisms contribute to improve and reinforce the decoding capabilities of parity bits.

Simulations with video-surveillance sequences have shown that the addition of parity bits offers significant improvement compared to pure intra JPEG 2000 refresh. Hence, the parity bits provides a way to preserve high access flexibility while decreasing the transmission cost in terms of bandwidth compared to pure INTRA-based conditional replenishment solutions.

The performances of the parity-only replenishment framework encourages the extension of this work toward the more conventional context of distributed coding, in which an encoder with a limited complexity is a major motivation. With respect to this, the three mechanisms described above should be implemented at the decoder side only, without impacting negatively the system performances. This migration to the client is left for future research.

# Low complexity scalable video server 5

*The content of this chapter has been published in [19, 20], and is extended here to the case of parity replenishment.*

*In this chapter, we consider the practical deployment of the replenishment system described in Chapter 3 to serve a large number of heterogeneous clients while preserving an acceptable computational complexity. The main objective is thus to support low cost adaptation to user requests and resources.*

## 5.1 Introduction

As a key and crucial contribution, we demonstrate in this chapter that most of the computation needed to take the replenishment decisions can be performed off-line, without preventing the server to adapt its decisions to the actual transmission resources and to the semantic interest defined on-line by a particular user during the browsing session. In practice, all these pre-computed informations are gathered in a file, named *Rate-Distortion Index File* (RDIF) in the following.

The above statement has important and interesting practical consequences. In particular, it means that a single index file is pre-computed and exploited to cover multiple transmission scenarios like the ones presented in Section 3.2, each scenario representing a particular interest expressed by the user. It also implies that the proposed server naturally adapts to fluctuating and heterogeneous bandwidth conditions.

Two important characteristics of our conditional replenishment system described in Chapter 3 are at the root of this interactive low complexity video server.

Firstly, the replenishment framework circumvents the drawbacks of closed-loop prediction systems while preserving coding efficiency by combining multiple refreshment mechanisms and references. In our case, two refreshment mechanisms are considered: the first considers INTRA coding of the fresh information, while the second is based on parity bits, which relaxes the usual constraint of encoder-decoder tight synchronization. This is especially relevant when addressing heterogeneous clients dealing with different prediction references as well as transmissions in lossy environments. In such contexts, the JPEG 2000 INTRA replenishments and the soft decoding of parity bits increase the stability of the entire system.

Secondly, the framework is able to adapt the content to client needs and semantic interest, e.g. spatial regions of interest defined by the user. These constraints are exploited independently of the compression stage, which means that they can be provided a posteriori, at transmission time by each individual user.

## 5.2 Serving multiple heterogeneous clients: rate-distortion index file definition

As explained in Section 3.4, rate-distortion optimal allocation of transmission resources ends up in selecting for each user the optimal precinct replenishment decisions, based on the convex-hull sustaining the JPEG 2000, parity and reference RD points. We will show that most of the information required for this allocation process can be computed only once, and off-line.

First, we introduce some notations. As explained in Section 3.6, the parity quantization levels have been set to the same value as the optimal JPEG 2000 quantizations levels. Hence, we denote $\mathcal{Q}$ the set of JPEG 2000 and parity layers corresponding to these quantization levels and $q$ the layer indexes, with $q \in \mathcal{Q}$.

In a video streaming context, we consider the transmission of frame $t$, and denote $d^{k,\kappa}(i,t)$ to be the distortion measured when approximating the $i^{th}$ precinct of frame $t$, based on the $\kappa$ first layers of the corresponding precinct in frame $(t-k)$. The replenishment distortion of precinct $i$ at time $t$ with $q$ JPEG 2000 layers is denoted $d_J(i,t,q) = d^{0,q}(i,t)$ and the size in bytes of these $q$ JPEG 2000 packets is denoted $s_J(i,t,q)$.

When parity bits aim at rendering the precinct $i$ of frame $t$ by correcting the corresponding $\kappa$ first layers of the corresponding precinct bits in frame $(t-k)$ with a quantization level $q$, the distortion of this parity packet generated is denoted $d_P(i,t,q)$ and the size in bytes[1] of this packet is $s_P^{k,\kappa}(i,t,q)$.

In absence of refreshment, the reference distortion for precinct $i$ at time $t$ is denoted $d_{ref}(i,t)$.

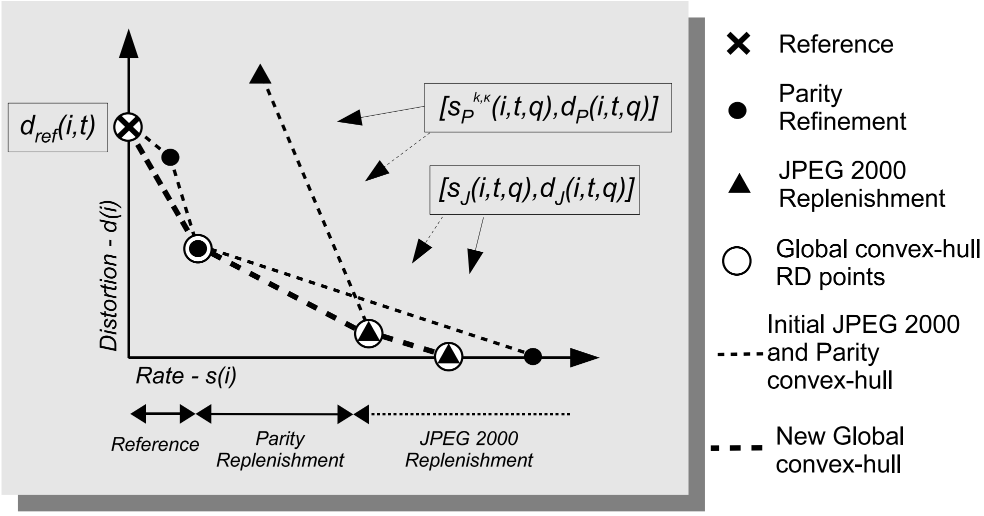


Figure 5.1: *Rate-distortion representation of the replenishment decisions for the $i^{th}$ precinct at time $t$. The variables required to compute the convex-hull for an optimal allocation are the reference distortion $d_{ref}(i,t)$ and the replenishment data lengths $s_J(i,t,q)$, $s_P^{k,\kappa}(i,t,q)$ and distortions $d_J(i,t,q)$ and $d_P(i,t,q)$. Note that for visual clarity, the quantization levels of JPEG 2000 and parity refreshments do not correspond in this figure.*

We reproduce a figure already presented in Section 3.4, which gives a rate-distortion representation of the possible replenishment decisions for a given precinct. From this figure, we observe that the following variables are required for the allocation process:

- $d_{ref}(i,t)$, the reference distortion for precinct $i$ at time $t$ in absence of parity or JPEG 2000 refreshment. Formally, when the latest replenishment of precinct $i$ occurred $k_t^i$ frames earlier than $t$ with a refresh of quality $\kappa_t^i$, the reference distortion $d_{ref}(i,t)$ for precinct $i$ at time $t$ is equal to $d^{k_t^i,\kappa_t^i}(i,t)$. When a background reference

[1] Section 4.5.2 details how this size is computed in practice.

is used, as in Section 3.5, the best replenishment solution between the previous replenishment reference and the background reference is calculated. Letting $b(i,t)$ denote the distortion obtained when approximating the $i^{th}$ precinct of frame $t$ based on the latest version of the background, the reference distortion is defined in this case by $d_{ref}(i,t) = min[d^{k_t^i,\kappa_t^i}(i,t)\ ,\ b(i,t)]$.

- $d_J(i,t,q)$, the distortion associated with a JPEG 2000 refreshment with layer $q$.

- $d_P(i,t,q)$, the distortion associated with a parity refreshment with layer $q$.

- $s_J(i,t,q)$, the size in bytes of a JPEG 2000 refreshment with layer $q$.

- $s_P^{k,\kappa}(i,t,q)$, the size in bytes of a parity refreshment with layer $q$ based on the $\kappa$ first layers of frame $(t-k)$.

The distortion and size in bytes of JPEG 2000 and parity refreshments must be computed for each JPEG 2000 and parity layer.

When the server has to cope with a large number of clients, possibly accessing distinct streams, the real-time calculation of the $s_P^{k,\kappa}(i,t,q)$ and $d_{ref}(i,t) = d^{k_t^i,\kappa_t^i}(i,t)\ \forall\ k > 0$ and $q \in \mathcal{Q}$ and $\kappa \in \mathcal{Q}$ values of interest becomes computationally intractable. In order to decrease this complexity, we propose to separate the process in two phases. During an off-line phase[1], the server performs once and for all most of the computationally expensive operations, and stores the results in an index file. This index is then exploited for on-line adaptive scheduling of packets, based on the actual resources and interest of a particular client.

Figure 5.2 depicts the off-line operations generating the compressed bitstreams and leading to the creation of the RDIF. The precincts issued from the wavelet transform are JPEG 2000 and parity encoded. The distortion and cost in bytes of the generated packets are passed to the RDIF. Similarly, the distortion of the reference is calculated and passed to the RDIF. To simplify, the generation of the background reference has not been depicted in this figure.

[1] Off-line means here that computations are performed independently of the actual semantic weights $w_t(i)$ or transmission resources experienced by a particular user.

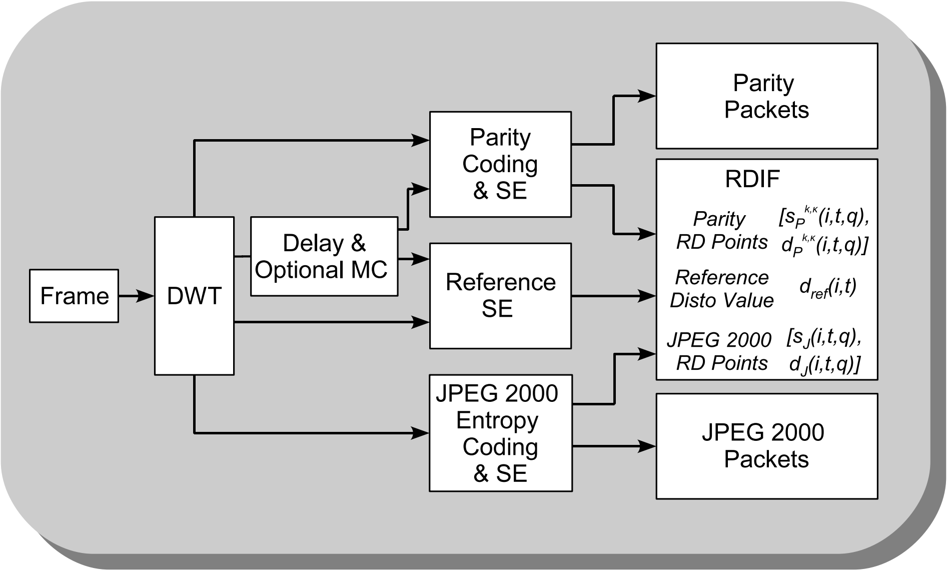


Figure 5.2: *Generation of compressed data and rate-distortion (RD) information at the encoder side. The main modules consist in the discrete wavelet transform (DWT), the delay and optional motion compensation module (MC), the generation of parity and JPEG 2000 data with their associated cost in bytes and squared error (SE), and the calculation of the reference squared error.*

### Reducing the cost associated to RD pre-computations

Among the values required for the allocation process, the JPEG 2000 distortion values $d_J(i,t,q)$ and byte costs $s_J(i,t,q)$ are directly computed during the encoding process, for each $q \in \mathcal{Q}$. For the parity distortion, this can be avoided since parity and JPEG 2000 layers quantization level correspond. Hence, $d_P(i,t,q) = d_J(i,t,q)$.

In contrast, the computation of the $d^{k,\kappa}(i,t)$ values which are used to calculate the reference distortion must be done for all $k > 0$ and $\kappa \in \mathcal{Q}$. Hence, this implies a significantly larger effort, both in terms of computation and memory resources. In the following section, we analyze closely the temporal evolution of these distortion values in order to approximate them with reduced complexity.

A similar problem arises for the parity replenishment lengths $s_P^{k,\kappa}(i,t,q)$. The number of parity bits required to correct a precinct at a given quantization level depends on the virtual channel noise[1] between the reference precinct of quality $\kappa$ located $k$ frames earlier and the current precinct. Since the computation of the virtual channel features for all layers and all possible previous frames is very high, an approximated estimation is proposed in Section 5.4, in which we also propose a method to adapt the length of parity packets as a function of this estimated virtual channel noise.

## 5.3 Estimation of the distortion induced by a previous reference replenishment

In this section, we explain how to estimate $d^{k,\kappa}(i,t)$ at low computational cost. To understand how the distortion of a precinct is affected by the time elapsed between the last refreshment and the current time, we have calculated the temporal evolution of the SE distortion between a precinct and its corresponding references in previous frames (see Figure 5.3), for the ten first frames of the *Speedway* sequence. The resulting distortions are illustrated in Figure 5.4.

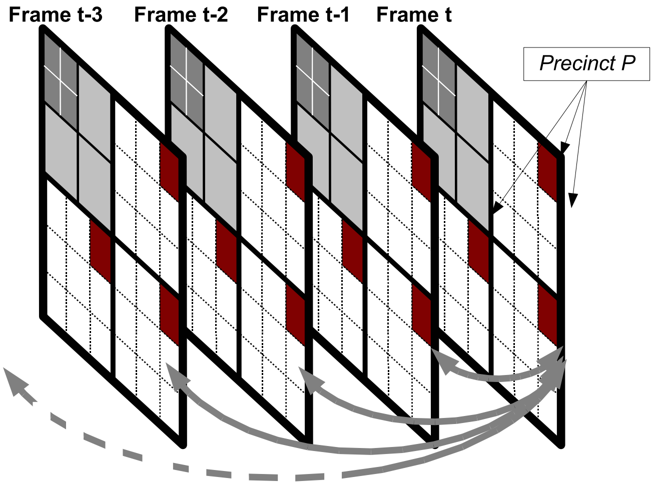
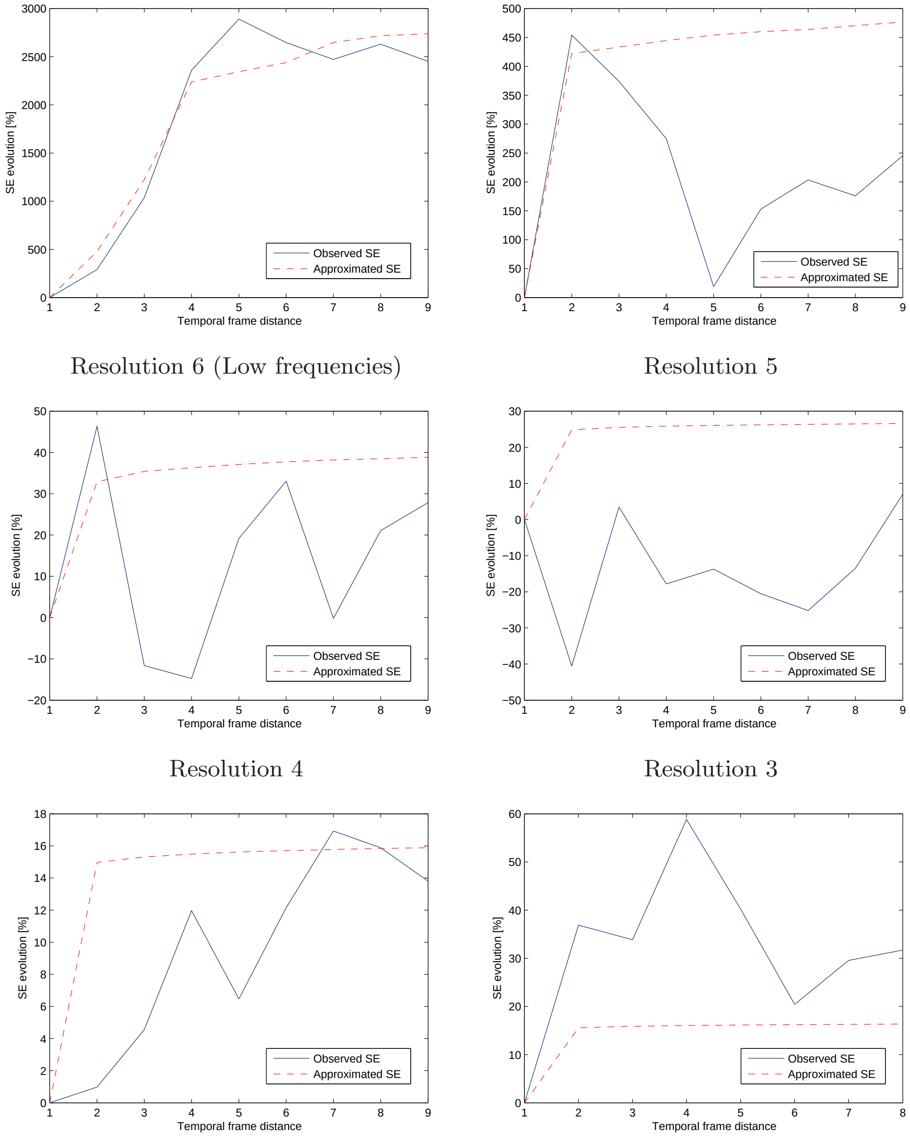


Figure 5.3: *Calculation of the temporal evolution of precinct $P$, which corresponds to the red zones of resolution 1.*

We first observe on Figure 5.4 that the SE increases with the temporal distance much more significantly for the low frequency resolutions than for

[1] The concept of virtual channel has been presented in Section 2.6



Resolution 6 (Low frequencies)

Resolution 5

Resolution 4

Resolution 3

Resolution 2

Resolution 1 (High frequencies)

Figure 5.4: *Temporal evolution of the distortion and approximated distortion between a precinct to transmit and the corresponding precincts in previous frames, for all resolutions. The values represent the SE increase in percents compared to the first frame SE. Note that the scale of the ordinates decreases significantly with the resolution.*

high frequency resolutions. A second observation is related to the temporal duration of the SE increase. In the lowest frequency resolution, the SE increases consistently during several frames while this is not the case for the higher frequencies.

These two observations can be explained by the fact that the sequence temporal correlation is mostly present in the low frequencies. Higher frequencies, which mostly contain sequence details and noise have a significantly lower temporal correlation. Hence, the SE evolution is more correlated to the frame distance in the low frequency resolutions, which makes it more predictable. This is important for the rate allocation process since errors on the SE prediction of low frequencies have a higher impact on the final quality than errors on the high frequencies[1].

Our SE approximation integrates these two observations and estimates the SE between distant frames by a weighted sum of the SE between consecutive intermediary frames. This approximation is motivated by Figure 5.5 which depicts the hierarchy of layers associated to frames between time $t-k$ and $t$. We denote $\kappa_{max}$ the quality layer with the highest quality. The distortion $d^{k,\kappa}(i,t)$ (dashed arrow in the figure) can be approximated based on a distortion computation path that only relies on $d_J(i,t-k,q)$ and $d^{1,\kappa_{max}}(i,\tau)$ values, with $t-k<\tau\leq t$, each step being characterized by a weight that depends on the precinct resolution and frame distance.

Formally, the equation corresponding to this approximation is the following:

$$\widehat{d^{k,\kappa}}(i,t) = d_J(i,t-k,\kappa) + \sum_{l=0}^{k-1} \omega(l,r)\; d^{1,\kappa_{max}}(i,t-l) \tag{5.1}$$

The role of the $\omega(l,r)$ term is to adapt the influence of the frame distance $l$ to the precinct resolution $r$, according to both previous observations.

$$\omega(l,r) = \alpha(r)\; e^{-(R-r)}\; e^{-l} \tag{5.2}$$

where $R$ corresponds to the total number of resolutions, $r$ corresponds to the precinct resolution index and is numbered as previously in this work

[1] Indeed, low frequencies have a higher weight in the rate allocation process since their $\gamma_s$ L2-norm of the wavelet basis functions is significantly higher. Section 3.3.2 explains how this norm which is integrated in the distortion metric has an influence on the rate allocation process.

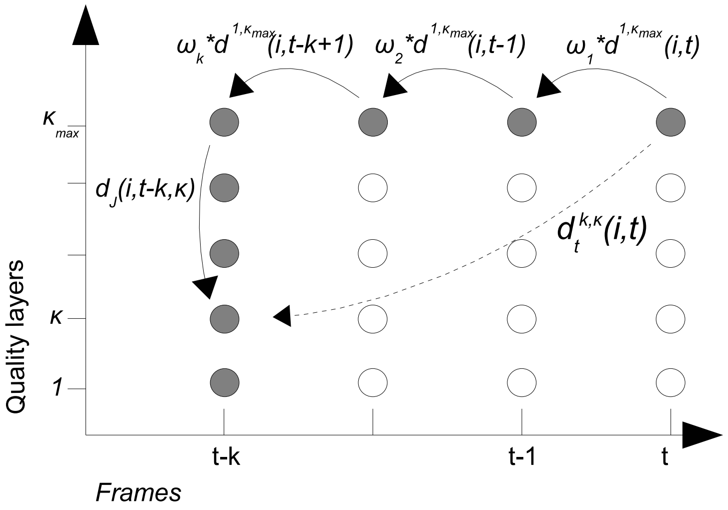


Figure 5.5: *Path used to approximate the distortion of the previous references, compared to the optimal path (dashed arrow). This approximation significantly decreases the pre-processing complexity and storage requirements, without significantly impairing the streaming performance.*

($r = R$ for the lowest resolution) and $\alpha(r)$ has been defined in our simulations as:

$$\alpha(r) = \frac{30}{1 + (R - r)} \tag{5.3}$$

Figure 5.4 illustrates the approximation (dashed red curve).

The gain in complexity resulting from this approximation is obvious: distortion values only need to be calculated between adjacent frames, at the highest quality level. For more distant frames, simple weighted sums of previously computed SE are necessary.

Formally, the distortion $d^{k,\kappa}(i,t)$ only relies on the one hand on $d_J(i, t-k, q)$ which corresponds to the distortion between layers in frame $t-k$ and is defined by the quantization levels of the encoding process, and on the other hand on $d^{1,\kappa_{max}}(i,\tau)$ values which are only calculated once for each frame. This significantly reduces the amount of values to compute and store in the index file, compared to $d_X^{Y,Q}(i)$, where $X$, $Y$ and $Q$ variables take all possible values.

If we denote $I_d$ to be the number of previous frames considered for the calculation of the previous reference distortion, we turn the $\mathbf{O}(I_d * \kappa_{max})$ complexity into $\mathbf{O}(I_d + \kappa_{max})$.

We will see below in Section 5.5 that this approximation does not have a significant impact on the system performances.

## 5.4 Parity coding rate allocation

We have explained in the previous section how to approximate at low computational cost the reduction of distortion resulting from the JPEG 2000 and parity replenishment options. We now consider the cost in bytes associated to these options.

In the case of JPEG 2000, the replenishment lengths $s_J(i, t, q)$ are computed during the off-line encoding process. For the parity replenishment lengths, the number of parity bits $s_P^{k,\kappa}(i, t, q)$ required to correct the reference depends on the virtual channel noise, which is specific to each client session since this reference depends on previous replenishment decisions. Hence, pre-computation of channel noise should consider all possible references for precinct $i$, ending in a tremendous computational work. We now explain how to reduce this computational load with minor impact on the RD optimal allocation process.

### 5.4.1 Virtual channel temporal evolution

In this work, as a first approximation, we characterize the virtual channel noise with the BER (Bit Error Rate). As it has been done previously with the SE, we first analyze how the BER evolves as a function of the temporal distance between the reference corrected based on parity bits, and the actual precinct to encode. This is illustrated in the Figure 5.6[1] which represents the BER of the four lower resolutions evolving with the frame distance.

[1] Only the four low resolutions are illustrated since the BER temporal evolution for higher resolutions is not significant. This can be explained by the fact that at high resolution, the temporal correlation between precincts is very low and hence the BER between these precincts is very high. This is true whether the precincts are temporally close or not.

The observations made in the previous section for the SE are valid for the BER. The BER increases are more significant and consistent with the temporal distance for low resolutions.

The proposed BER temporal approximation is very similar to the SE one. It also restricts the computation of BER values to adjacent frames and consists in a weighted sum of BER values computed on previous frames. Formally, we denote $BER^{k,\kappa}(i,t,q)$ to be the BER measured when approximating the $q^{th}$ layer of the $i^{th}$ precinct of frame $t$, based on the $\kappa$ first parity layers of the corresponding precinct in frame $(t-k)$. In particular, the replenishment of precinct $i$ at time $t$ with $q$ layers has an associated BER denoted $BER(i,t,q) = BER^{0,q}(i,t,q_{max})$.

Similarly to the SE approximation, $BER^{k,\kappa}(i,t,q)$ can be approximated based on a BER computation path that relies on $BER(i,t-k,\kappa)$ and $BER^{1,\kappa_{max}}(i,\tau,\kappa_{max})$ values, with $t-k < \tau \leq t$, each step characterized by a weight which depends on the precinct resolution and frame distance. However, since in this case we are calculating the BER between precincts that both can be encoded at any layer ($q$ and $k$), we add to the approximation the term $BER(i,t,q)$, which corresponds to the BER between between layer $q_{max}$ and $q$ of the $i^{th}$ precinct at time $t$.

Formally, the equation corresponding to this approximation is the following:

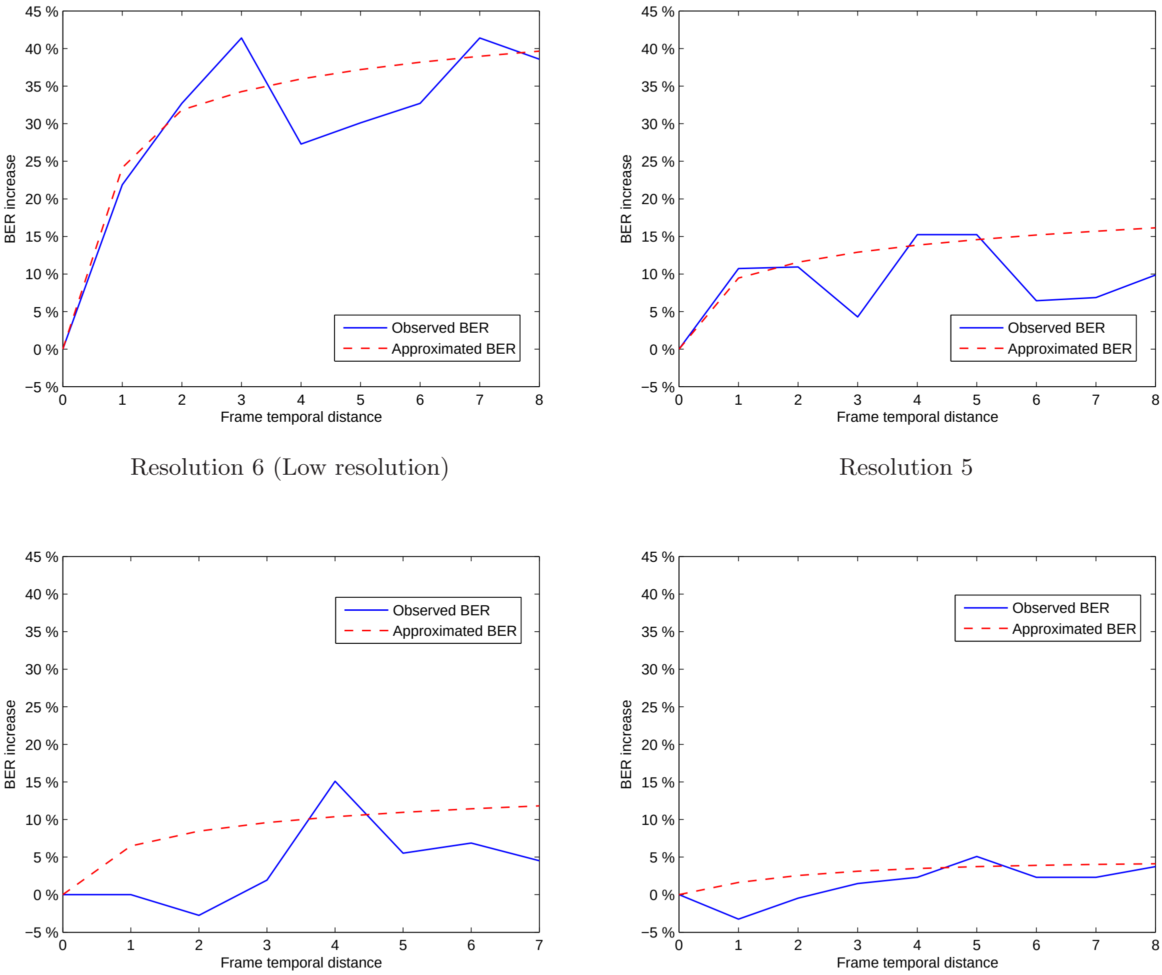
$$\widehat{BER}^{k,\kappa}(i,t,q) = BER(i,t-k,\kappa) + \sum_{l=0}^{k-1} \omega'(l,r)\; BER^{1,\kappa_{max}}(i,t-l,\kappa_{max}) + BER(i,t,q) \quad (5.4)$$

where $\omega'(l,r)$ is defined as

$$\omega'(l,r) = e^{-\beta_1\, l}\; e^{-\beta_2\,(R-r)} \quad (5.5)$$

where $\beta_1 = 0.01$, $\beta_2 = 0.5$ in our simulations, and $R$ corresponds to the total number of resolutions and $r$ to the precinct resolution index and is numbered as previously in this work ($r = R$ for the lowest resolution).

Like in the SE case, BER values only need to be calculated between adjacent frames, at the highest quality level. For more distant frames, simple weighted sums of previously computed BER are necessary. Formally, the



Resolution 6 (Low resolution)

Resolution 5

Resolution 4

Resolution 3

Figure 5.6: *BER between a precinct to transmit and the corresponding precincts in previous frames, for the four lower resolutions, as a function of the temporal distance between both frames. The values represent the percentage of BER increase compared with the first frame BER. The proposed approximation for these BER is represented in dashed red curves.*

distortion $BER^{k,\kappa}(i,t)$ only relies on $BER(i,t-k,\kappa)$ which corresponds to the BER between layers in frame $t-k$, on $BER^{1,\kappa_{max}}(i,\tau)$ values which are only calculated once for each frame and on $BER(i,t,q)$. This significantly reduces the amount of values to compute and store in the index file, compared to $BER_X^{Y,Q}(i)$, where $X$, $Y$ and $Q$ variables take all possible values. Again, if we denote $I_{BER}$ to be the number of previous frames considered for the calculation of the BER, we turn the $\mathbf{O}(I_{BER} * \kappa_{max})$ complexity into $\mathbf{O}(I_{BER} + \kappa_{max})$.

### 5.4.2 Modelization of the evolution of the parity length with the virtual channel noise

Now that the scheduler can exploit a low complexity approximation to estimate the BER between a precinct and its reference, it still requires to estimate the number of parity bits to transmit in order to correct the reference precinct, based on this estimated BER.

Hence, we now have to calculate $s_P^{k,\kappa}(i,t,q)$, the minimal parity length required to correct the reference. Practically, a limited number of LDPC matrices $\mathbf{H}$ have been generated, as explained in Section 4.5. The cost in bytes $s_P^{k,\kappa}(i,t,q)$ corresponds to the codeword length associated with the smallest of these matrices able to correct the reference.

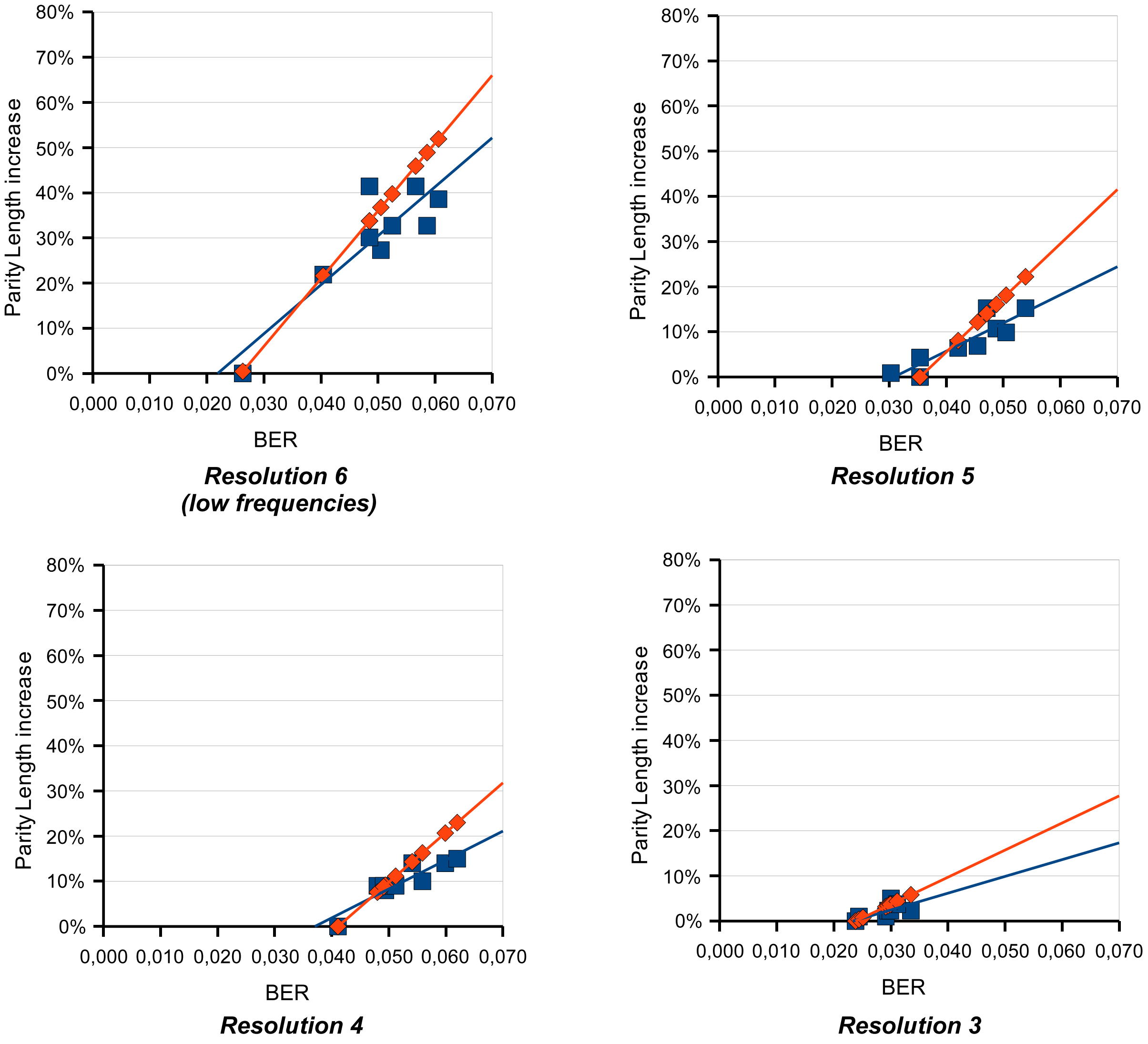
Figure 5.7 represents the evolution of $s_P^{k,\kappa}(i,t,q)$ as a function of the various BER observed with the previous references. More precisely, the $X$ axis represents the BER observed between precinct $i$ at time $t$ and the same precinct at time $(t-k)$, for $1 \leq k \leq 9$, and the $Y$ axis represents the increase of $s_P^{k,\kappa}(i,t,q)$ compared to $s_P^{1,q}(i,t,q)$. The observed values are represented by blue dots and the proposed approximation by red dots.

Formally, the estimated parity packet length is:

$$\widehat{s_P^{k,\kappa}}(i,t,q) = s_P^{1,q}(i,t,q) + \mu(r) * (BER^{k,\kappa}(i,t,q) - (BER^{1,q}(i,t,q)\ ) + \nu(r) \tag{5.6}$$

where $\mu(r)$ and $\nu(r)$ denotes the parameters of the linear approximation. These parameters are calculated once for each resolution $r$. Figure 5.8 illustrates this approximation. The $s_t^{1,q_P}(i)$ values are stored in the RDIF, and the BER values are approximated as explained in the previous section.

Regarding Figure 5.7, we observe as expected that $s_t^{k,q_P}(i)$ increases when the noise increases, and as discussed previously, the BER variations



**Resolution 6 (low frequencies)**

**Resolution 5**

**Resolution 4**

**Resolution 3**

Figure 5.7: *Evolution of the parity length with the virtual channel noise, for the four lower resolutions. The blue dots represent the observed values, the red dots the proposed approximation and the two lines represent a linear regression of both sets.*

- and hence the parity length variations - are more significant in the low resolutions. The parity length approximations are usually slightly above

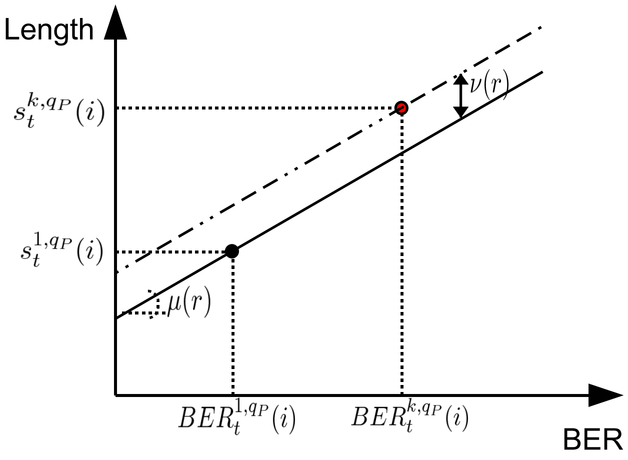


Figure 5.8: *Illustration of the approximation used to estimate the parity packet length $s_t^{k,q_P}(i)$. It is based on the $s_t^{1,q_P}(i)$ value available in the RDIF and BER approximations presented in Section 5.4.1.*

the required lengths, i.e. the blue dots are usually located above the red ones, which has a (small) impact on the parity performances. In few cases, the approximation is below the required length. In this case, the decoder will not be able to correct the reference. The two parameters $\mu(r)$ and $\nu(r)$ can decrease this number of uncorrectable references, at the expense of a slight decrease in performances.

## 5.5 Results

In this section, we first analyze the impact of proposed temporal distortion and parity length approximations on the quality of transmitted sequences at various bitrates. We then quantify the gain in memory and computational resources that obtained with these approximations for a specific transmission scenario.

### 5.5.1 Transmission quality

Figure 5.9 and Figure 5.10 compare the performances of the optimal transmission system (without approximations) with the system integrating the SE approximations described in Section 5.3 for the *Speedway* and *Caviar* sequences respectively, using the JRB method (see Section 3.7.2). We observe that the system performances are penalized at low rates, but rapidly reach those of the optimal method as the rates increase.

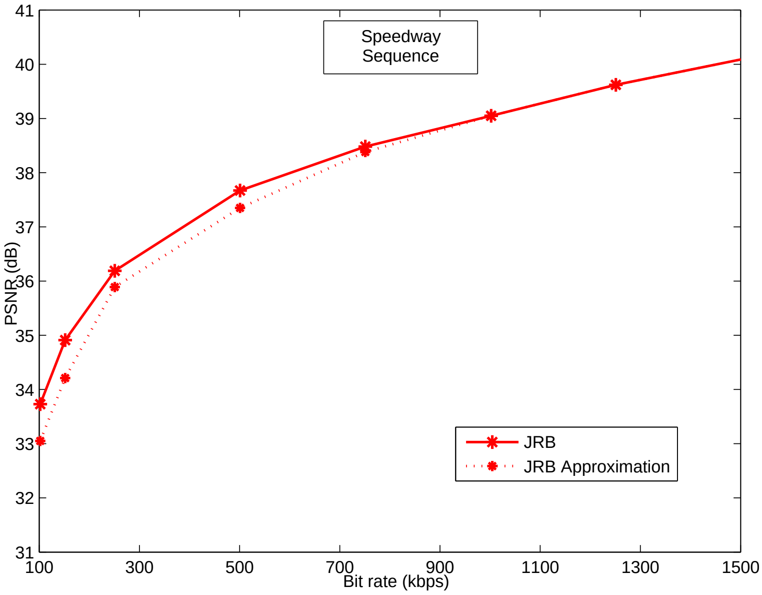


Figure 5.9: *Comparison between the transmission of the Speedway sequence with and without the temporal distortion approximation.*

Figure 5.11 compares the performances of the parity rate allocation with and without the parity length approximations presented in Section 5.4 for

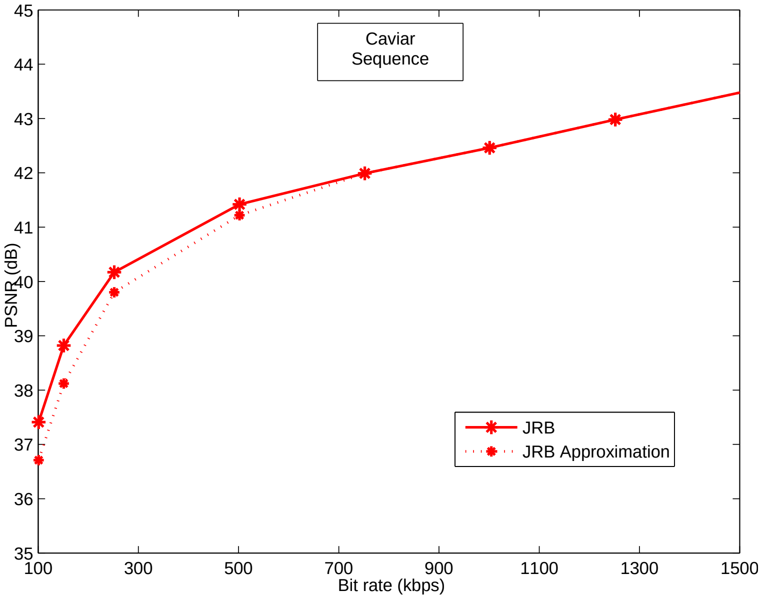


Figure 5.10: *Comparison between the transmission of the Caviar sequence with and without the temporal distortion approximation.*

the PR method (See Section 3.7.1). We observe that these approximations have a slightly more significant impact as the rate increases. This is explained by the fact that at higher rates, a larger number of parity packets must be transmitted, increasing the parity length approximation errors and hence increasing the number of unnecessary parity bits transmitted.

### 5.5.2 Memory and computational resources

The gain in memory and computational resources enabled by the proposed approximations is illustrated in Figure 5.12, which plots the memory and computational complexity requirements for three different implementations of our proposed replenishment framework within a video server. For clarity, we only consider the JR replenishment method to illustrate this gain and hence focus on the temporal distortion approximations, but similar gains are of course observed for the JPR method which can combine temporal distortion approximations with parity length approximations.

The first implementation, called *Optimal Online* strategy, computes the reference distortion required for the rate allocation of each individual user

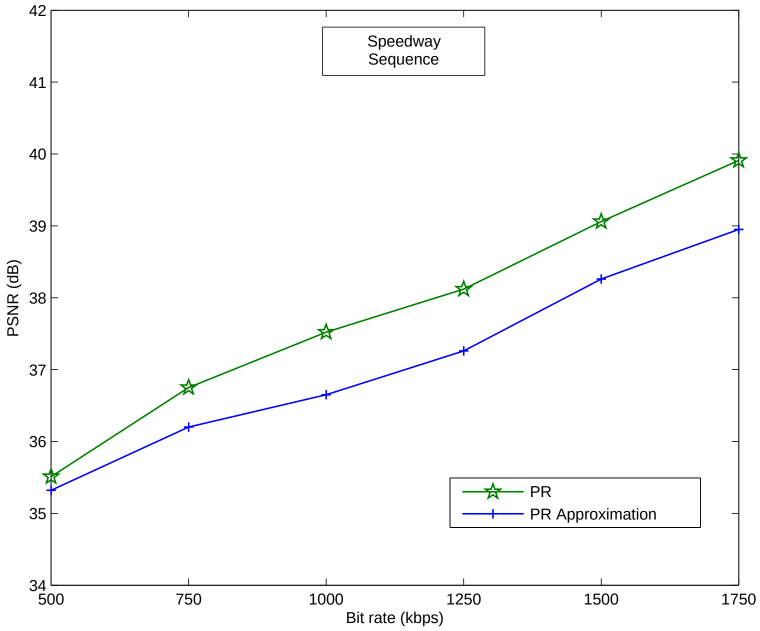
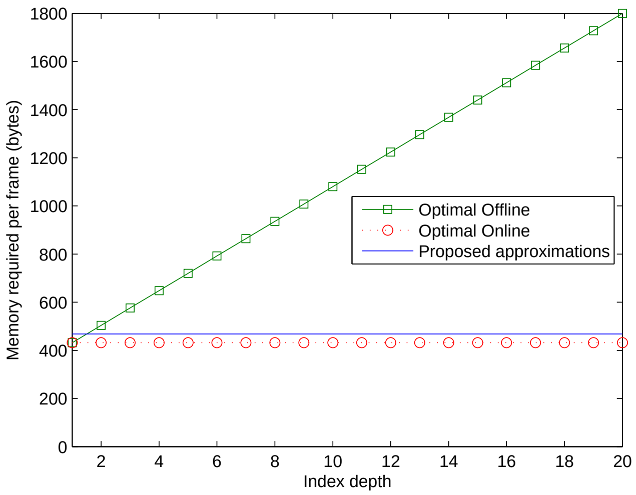


Figure 5.11: *Impact of the parity length approximations on the received sequence quality. The results are obtained using the PR method for the Speedway Sequence.*

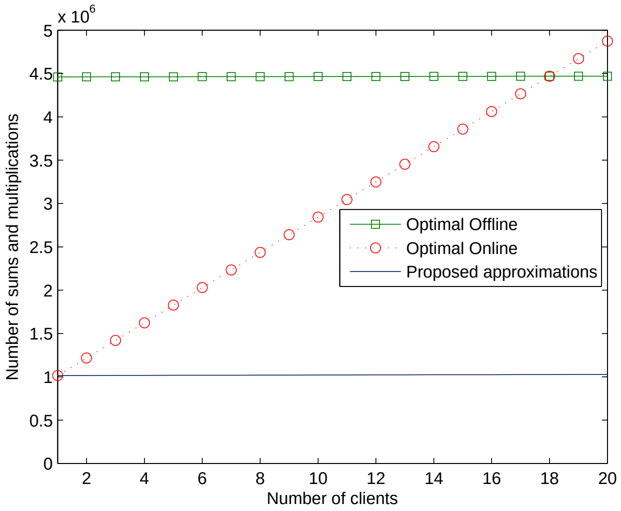
at transmission time, knowing the exact scheduling history of the user. The *Optimal Offline* strategy computes and stores all possible reference distortions off-line, without any approximation, by anticipating all the precinct references resulting from possible earlier transmission strategies. The *Proposed approximations* strategy only pre-computes the distortion resulting from the approximation of a precinct based on its previous correspondence, encoded at the highest quality level and uses Equation 5.1 to estimate the missing reference distortions, as explained in Section 5.3.

In Figure 5.12(a), we observe that the memory required by the *Optimal Offline* strategy increases linearly with $I_d$[1], while it remains constant for the other strategies. In Figure 5.12(b), the server complexity is measured in terms of the numbers of arithmetic operations required for the computation of the distortion with $I_d$ set to 10. We observe that the complexity of the *Optimal Online* strategy increases significantly with the number of users,

[1] Recall that $I_d$ corresponds to the number of previous frames considered for the calculation of the previous reference distortion



$(a)$



$(b)$

Figure 5.12: *Comparison of (a) memory and (b) computational resources for three different implementations of the proposed replenishment framework. Memory is depicted as a function of the index depth $I_d$, while computational resources are presented as a function of the number of server clients.*

since the computation are performed independently for each client. For the other strategies, most operations are performed offline, and the additional required online operations appear to be insignificant compared to the offline operations. This is reflected by nearly horizontal curves for these strategies in Figure 5.12(b).

## 5.6 Conclusion

In this chapter, we have presented a low complexity server based on the conditional replenishment mechanisms presented in Chapter 3. This server is able to adapt in real-time and low complexity a single pre-encoded content to possibly a very large number of heterogeneous users with different needs and interest in the content.

We have shown that this scheduling is based on a light rate allocation, which relies on approximations based on pre-computed values. These values are generated once during an off-line phase and store in an index file. We have shown that these approximations affect the system performances in an acceptable way, while significantly decreasing the computation complexity and required memory for the off-line generation of the index file.

# Conclusions 6

*In this thesis we have addressed the problematic of an efficient and flexible remote browsing of video content. The proposed system offers fine-grained scalability in terms of resolution, quality and spatial access as well as temporal access to individual frames and enables users with very different profiles, resources and interests to access efficiently the stored content.*

*In the present conclusion, we summarize the main contributions of this work and present the perspectives and future directions for this research.*

## 6.1 Contributions

As a first contribution, we have liberated the coding framework from the strict closed-loop prediction required in conventional hybrid video coding schemes. Intra-based and parity-based replenishment solutions have been considered to increase robustness to a mismatch between the references available at the encoder and the decoder. On the one hand, intra content opens and virtually removes the prediction-loop refreshing the content. On the other hand, parity bits are designed to correct stochastic errors, and not to encode deterministic prediction errors. Hence, the system is expected to support some desynchronization between the encoder and decoder, which is known to be particularly helpful when the content is pre-encoded off-line, and the transmission server has to adapt to fluctuating and not guaranteed network resources.

As a second contribution, we have proposed a rate-distortion optimal strategy to select the most profitable data to transmit among multiple replenishment options, thereby unifying open-loop (JPEG 2000 INTRA) and relaxed closed-loop (parity bits) mechanisms. This rate allocation is independent of the compression engine, which enables the server to adapt in real-time the content forwarded to heterogeneous - both in terms of resources and interest - clients using a single pre-compressed version of the sequence. A special attention has been payed to the complexity reduction of this scheduling, enabling the system to be scaled for serving a very large number of sequences and users. The result consists in a worthy compromise between coding efficiency and server complexity.

As a third and significant outcome, we have integrated the novel technique of coding with side information into the conventional conditional replenishment framework. To preserve scalability, the side information has been exploited on bit-planes, in the wavelet transform domain. Hence, a particular attention has been devoted to the efficient exploitation of temporal but also spatial correlation among wavelet subbands coefficients, while defining the parity bits on subsets of wavelet bit-planes to preserve quality scalability. A particular attention has been devoted to understanding how an image source model can improve parity bits correcting capabilities. In that context, we have observed that formalizing the spatial correlation between coefficients mainly helps in presence of localized errors. This finding is important since it should drive the design of the motion-compensation stage that should be envisioned for extending our system to arbitrary moving video content.

## 6.2 Perspectives

As explained above, a first direction for future work is the design of a motion estimation module designed in a complementary way to the parity coding spatial modeler. This work includes the research of new optimization parameters specific to the parity replenishment.

In the present work, regular LDPC transformation matrices have been used for the parity replenishment. A deep analysis of the particular virtual channel present in this replenishment system combined with the research of irregular matrices more adapted to this particular channel deserves further investigation.

A paradigm shift consisting in reducing the scalability constraint, and instead envisioning a low complexity encoder is also an interesting path. This work is more in phase with the conventional Distributed Video Coding (DVC) research which is nowadays very active. This low complexity encoder would only integrate the parity replenishment combined with a rate allocation using the distortion and virtual channel approximations presented in this work. Since complexity is shifted to the decoder in the DVC paradigm, most of the temporal and spatial correlation should be exploited after the reception. Should motion estimation be considered, it would as be done at the decoder side.

Finally, since an interesting feature of the proposed framework is its robustness, transmission in more hostile environments should be considered, extending the work presented in Section 3.7.3. In this case, the physical transmission channel (as opposed to the virtual channel referred to above) conditions should be integrated during the rate allocation stage. Clearly, parity bit replenishments will be favored as they will be able to correct both virtual and physical channels. This should be achieved without significantly increasing the system complexity or developing new modules. The research presented in [7] which proposes such allocation process adapted for noisy channels in a JPEG 2000 transmission context could provide a starting point for this study.

# Publications

## Journal papers

- F.-O. Devaux, and C. De Vleeschouwer. Parity bit replenishment for JPEG 2000 based video coding. Accepted under revisions to *EURASIP, Journal on Image and Video Processing, Special issue on Distributed Video Coding, April 2008.*
- F.-O. Devaux, C. De Vleeschouwer, J. Meessen, C. Parisot, B. Macq, and J.-F. Delaigle. Remote interactive browsing of video surveillance content based on JPEG 2000. Accepted under revisions to *IEEE Transactions on Circuits and Systems for Video Technology, 2008.*
- M. Agueh, F.-O. Devaux, M. Diop, J.-F. Diouris, C. De Vleeschouwer and B. Macq. Optimal Wireless JPEG 2000 compliant Forward Error Correction rate allocation for robust JPEG 2000 images and video streaming over Mobile Ad-hoc Networks. *EURASIP, Journal on advances in Signal Processing, 2008.*
- A. Descampe, F.-O. Devaux, G. Rouvroy, J.-D. Legat, J.-J. Quisquater, and B. Macq. A Flexible Hardware JPEG 2000 Decoder for Digital Cinema. *IEEE Trans. on Circuits and Systems for Video Technology*, 16(11):1397-1410, November 2006.

## Conference papers

- A. Massoudi, F. Lefebvre, C. De Vleeschouwer and F.-O. Devaux. Secure and low cost selective encryption for JPEG 2000. In *Proceedings of the 10th IEEE International Symposium on Multimedia (ISM-2008)*, December 2008.
- F.-O. Devaux, L. Schumacher and C. De Vleeschouwer. A flexible video server based on a low complex post-compression rate allocation. In *Proceedings of the 16th International Packet Video Workshop (PV-2007)*, November 2007.

- Max Agueh, F.-O. Devaux, M. Diop, and J.F. Diouris. Dynamic channel coding for efficient Motion JPEG2000 video streaming over Mobile Ad-hoc Networks. In *Proceedings of the Third International Mobile Multimedia Communications Conference (MobiMedia-2007)*, August 2007.

- M. Agueh, F.-O. Devaux and J.F Diouris. A Wireless Motion JPEG 2000 video streaming scheme with a priori channel coding. In *Proceedings of the 13th European Wireless 2007 (EW-2007)*, April 2007.

- F.-O. Devaux, J. Meessen, C. Parisot, J.-F. Delaigle, B. Macq, and C. De Vleeschouwer. A Flexible Video Transmission System based on JPEG 2000 Conditional Replenishment with Multiple References. In *Proceedings of the 32nd International Conference on Acoustics, Speech, and Signal Processing (ICASSP-2007)*, April 2007.

- A. Descampe, F.-O. Devaux, G. Rouvroy, J.-D. Legat, and B. Macq. An Efficient FPGA Implementation of a Flexible JPEG 2000 Decoder for Digital Cinema. In *Proceedings of the 12th European Signal Processing Conference (EUSIPCO-2004)*, September 2004.

- A. Descampe and F.-O. Devaux. A Flexible, Line-Based, JPEG 2000 Decoder for Digital Cinema. In *Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference (MELECON-2004)*, May 2004.

# Bibliography

[1] ThreePastShop1front sequence from the CAVIAR Project (Context Aware Vision using Image-based Active Recognition). http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1, 2001. 53

[2] FP6 IST-2003-507204 WCAM, Wireless Cameras and Audio-Visual Seamless Networking, 2004. WCAM Project website, hosting the Speedway Sequence: http://www.ist-wcam.org. 53

[3] ISO/IEC 15444-1. JPEG 2000 image coding system, 2000. 2, 10, 14, 38, 39, 80, 86, 92

[4] A. Cavallaro, O. Steiger and T. Ebrahimi. Semantic video analysis for adaptive content delivery and automatic description. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(10):1200–1209, October 2005. 40

[5] A. Aaron and B. Girod. Compression with side information using turbo codes. *Proceedings of the Data Compression Conference*, pages 252–261, 2002. 23

[6] A. Aaron, S. Rane, E. Setton, and B. Girod. Transform-Domain Wyner-Ziv Codec for Video. In *SPIE Visual Communications and Image Processing Conference*, San Jose, California, USA, January 2004. 23

[7] M. Agueh, F.-O. Devaux, M. Diop, J.-F. Diouris, C. De Vleeschouwer, and B. Macq. Optimal Wireless JPEG 2000 compliant Forward Error Correction rate allocation for robust JPEG 2000 images and video streaming over Mobile Ad-hoc Networks. *EURASIP, Journal on advances in Signal Processing*, 2008. 119

[8] M. Agueh, F.O. Devaux, M. Diop, J.-F. Diouris, C. De Vleeschouwer, and B. Macq. Optimal Wireless JPEG 2000 compliant Forward Error Correction rate allocation for robust JPEG 2000 images and video

streaming over Mobile Ad-hoc Networks. *EURASIP, Journal on advances in Signal Processing*, 2008. 68

[9] S. Andriani, G. Calvagno, T. Erseghe, GA Mian, M. Durigon, R. Rinaldo, M. Knee, P. Walland, and M. Koppetz. Comparison of lossy to lossless compression techniques for digital cinema. *Image Processing, 2004. ICIP'04. 2004 International Conference on*, 1, 2004. 32

[10] C. Berrou and A. Glavieux. Near optimum error correcting coding and decoding: Turbo-codes. *IEEE Transactions on Communications*, 44: 1261–1271, 1996. 75

[11] JR Casas and L. Torres. A region-based subband coding scheme. *Signal Processing: Image Communication*, 10(1):173–200, 1997. 40

[12] N.M. Cheung and A. Ortega. Compression algorithms for flexible video decoding. *Proceedings of SPIE*, 2008. 33

[13] C.A. Cole, S.G. Wilson, E.K. Hall, and T.R. Giallorenzi. A General Method for Finding Low Error Rates of LDPC Codes. *Arxiv preprint cs/0605051*, 2006. 84

[14] D. Santa-Cruz and T. Ebrahimi. An analytical study of JPEG 2000 functionalities. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Vancouver, September 2000. 2

[15] D. Taubman and R. Rosenbaum. Rate-distortion optimized interactive browsing of JPEG 2000 images. In *IEEE International Conference on Image Processing (ICIP)*, September 2003. 14, 45, 51

[16] D. Taubman D. and M. Marcellin. *JPEG 2000: Image compression fundamentals, standards and practice.* Kluwer Academic Publishers, 2001. 10

[17] A. Descampe, F.-O. Devaux, G. Rouvroy, J.-D. Legat, J.-J. Quisquater, and B. Macq. A Flexible Hardware JPEG 2000 Decoder for Digital Cinema. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(11):1397–1410, 2006. 92

[18] F.-O. Devaux and C. De Vleeschouwer. Parity bit replenishment for JPEG 2000 based video coding. *Submitted to EURASIP, Journal on Image and Video Processing, available at http://www.tele.ucl.ac.be/˜devaux*, 2008. 73

[19] F.-O. Devaux, L. Schumacher, and C. De Vleeschouwer. A flexible video server based on a low complex post-compression rate allocation. In *Proceedings of 16th IEEE International Packet Video Workshop*, Lausanne, Switzerland, November 2007. 97

[20] F.-O. Devaux, C. De Vleeschouwer, J. Meessen, C. Parisot, B. Macq, and J.-F Delaigle. Remote interactive browsing of video surveillance content based on JPEG 2000. *Submitted to IEEE Transactions on Circuits and Systems for Video Technology, available at http://www.tele.ucl.ac.be/˜devaux*, 2008. 31, 97

[21] F.-O. Devaux, J. Meessen, C. Parisot, J.F. Delaigle, B. Macq and C. De Vleeschouwer. A flexible video transmission system based on JPEG 2000 conditional replenishment with multiple references. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 07)*, Hawaii, USA, April 2007. 31

[22] M. Fresia and G. Caire. A practical approach to lossy joint source-channel coding. *Submitted to IEEE Transactions On Information Theory*, 2007. 68, 93

[23] R.G. Gallager. Low Density Parity Check Codes. *Number 21 in Research monograph series. MIT Press, Cambridge, Mass.*, 1963. 75

[24] J. Gantz, D. Reinsel, C. Chute, W. Schlichting, J. McArthur, S. Minton, I. Xheneti, A. Toncheva, and A. Manfrediz. The expanding digital universe, a forecast of worldwide information growth through 2010. In *IDC white paper*, March 2007. 1

[25] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack problems*. Springer Verlag, 2004. ISBN 3-540-40286-1. 42

[26] S.-C Han and J.W. Woods. Adaptative Coding of Moving Objects for Very Low Bit Rates. *IEEE Journal on Selected Areas in Communications*, 16(1):56–70, January 1998. 40

[27] ISO/IEC. International standard 15444, Information technology - JPEG 2000 image coding system, particularly Part 3: Motion JPEG 2000 (September 2002, with subsequent amendments). 54

[28] ISO/IEC JTC 1. “Coding of audio-visual objects - Part 2: Visual,” ISO/IEC 14492-2 (MPEG-4 Visual), Version 1: Apr. 1999, Version 2: Feb. 2000, Version 3: May 2004. 9

[29] ITU-T . "Video coding for low bit rate communication," ITU-T Recommendation H.263, Version 1: Nov. 1995, Version 2: Jan. 1998, Version 3: Nov. 2000. 9

[30] ITU-T and ISO/IEC JTC 1. "Advanced video coding for generic audio-visual services," ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), Version 1: May 2003, Version 2: May 2004, Version 3: Mar. 2005, Version 4: Sep. 2005, Version 5 and Version 6: June 2006, Version 7: Apr. 2007, Version 8 (including SVC extension): Consented in July 2007. 10, 15

[31] ITU-T and ISOIEC JTC 1. "Generic coding of moving pictures and associated audio information - Part 2: Video," ITU-T Recommendation H.262 and ISO/IEC 13818-2 (MPEG-2 Video), Nov. 1994. 9

[32] J. Meessen, C. Parisot, X. Desurmont and J.F. Delaigle. Scene Analysis for Reducing Motion JPEG 2000 video Surveillance Delivery Bandwidth and Complexity. In *IEEE International Conference on Image Processing (ICIP 05)*, volume 1, pages 577–580, Genova, Italy, September 2005. 50

[33] H. W. Jones. A conditional replenishment hadamard video compressor. In *Proceedings of the International Optical Computing Conference*, pages 91–98, San Diego, USA, August 1977. 18

[34] JPEG committee. Scope and Requirements for Advanced Image Coding (AIC) version 2.0, ISO/IEC JTC 1/SC 29/WG1 N3914. March 2006. 15

[35] J.L. Kim, UN Peled, I. Perepelitsa, V. Pless, and S. Friedland. Explicit construction of families of LDPC codes with no 4-cycles. *IEEE Transactions on Information Theory*, 50(10):2378–2388, 2004. 84

[36] K. Kim, T.H. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground–background segmentation using codebook model. *Real-Time Imaging*, 11(3):172–185, 2005. 47

[37] R. Koenen. Overview of the MPEG-4 Standard. *ISO/IEC JTC1/SC29/WG11 N*, 2001. 40

[38] F.R. Kschischang, B.J. Frey, and H.A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001. 75, 76

[39] L. Wolsey. *Integer Programming.* Wiley, 1998. 42

[40] M. Luby. LT codes. *Proceedings of the 43rd Annual IEEE Symposium on foundations of Computer Science, 2002*, pages 271–280, 2002. 84

[41] M. Rabbani and R. Joshi. An overview of the JPEG 2000 image compression standard. *Signal Processing: Image processing*, 17:3–48, 2002. 2, 10

[42] DJC MacKay. Good error-correcting codes based on very sparse matrices. *IEEE Transactions on Information Theory*, 45(2):399–431, 1999. 84

[43] DJC MacKay. Fountain codes. *IEE Communications*, 152(6):1062–1068, 2005. 84

[44] D. Marpe, V. George, H.L. Cycon, and K.U. Barthel. Performance evaluation of Motion-JPEG 2000 in comparison with H. 264/AVC operated in pure intracoding mode. *Proceedings of SPIE*, 5266:129–137, 2003. 32

[45] A Mavlankar, D. Varodayan, and B. Girod. Region-of-interest prediction for interactively streaming regions of high resolution video. In *Proceedings of 16th IEEE International Packet Video Workshop (PV)*, Lausanne, Switzerland, November 2007. 37

[46] S. McCanne, M. Vetterli, and V. Jacobson. Low-complexity video coding for receiver-driven layered multicast. *IEEE Journal on Selected Areas in Communications*, 15(6):983–1001, 1997. 18

[47] Steven McCanne and Van Jacobson. vic: A flexible framework for packet video. In *ACM Multimedia*, pages 511–522, 1995. 18

[48] J. L. Mitchell and W. B. Pennebaker. Software implementations of the Q-coder. *IBM J. Res. Develop.*, 32(6):753–774, November 1988. 12

[49] FW Mounts. A video encoding system with conditional picture-element replenishment. *Bell Systems Technical Journal*, 48(7):2545–2554, 1969. 18

[50] MPEG and ITU-T. Scalable Video Coding Standard ISO/IEC 14496-10. August 2007. 36

[51] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG. Joint Final Commitee Draft (JFCD) of Joint Video Specification (ITU-T Rec. H.264 – ISO/IEC 14496-10 AVC). Doc. JVT-D157, July 2002. 4

[52] E. Ordentlich, M. Weinberger, and G. Seroussi. A low-complexity modeling approach for embedded coding of wavelet coefficients. *Proceedings of the Data Compression Conference*, pages 408–417, 1998. 11

[53] A. Ortega, K. Ramchandran, and M. Vetterli. Optimal trellis-based buffered compression and fast approximation. *IEEE Transactions on Image Processing*, 3(1):26–40, January 1994. 41

[54] Antonio Ortega. Optimal bit allocation under multiple rate constraints. In *Data Compression Conference*, pages 349–358, Snowbird, UT, April 1996. 41

[55] A. Ortego and K. Ramchandran. Rate-distortion methods for image and video compression. *Signal Processing Magazine, IEEE*, 15(6):23–50, 1998. 41

[56] R. Puri and K. Ramchandran. PRISM: A new robust video coding architecture based on distributed compression principles. *Proceedings of the Allerton Conference on Communication, Control, and Computing*, 2002. 23

[57] S. Rane, A. Aaron, , and B. Girod. Systematic Lossy Forward Error Protection for Error-Resilient Digital Video Broadcasting. In *SPIE Visual Communications and Image Processing Conference*, San Jose, California, USA, January 2004. 23

[58] T.J. Richardson and R.L. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Transactions on Information Theory*, 47(2):599–618, 2001. 84

[59] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 2007 September. 15, 16, 17

[60] JM Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462, 1993. 13

[61] Y. Shoham and A. Gersho. Efficient bit allocation for an arbitrary set of quantizers. *IEEE Transactions on Signal Processing*, 36(9):1445–1453, September 1988. 41, 45

[62] A. Shokrollahi. Raptor codes. *IEEE/ACM Transactions on Networking (TON)*, 14:2551–2567, 2006. 84

[63] D. Slepian and J. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, 19(4):471–480, 1973. 21

[64] M. Smith and J. Villasenor. Intra-frame JPEG-2000 vs. Inter-frame Compression Comparison: The benefits and trade-offs for very high quality, high resolution sequences. *SMPTE Proceedings: Technical Conference and Exhibition*, pages 1–9, 2004. 32

[65] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 246–252, June 1999. 47

[66] T. Wiegand, G.J. Sullivan, G. Bjntegaard, A. Luthra. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, July 2003. 4

[67] D. Taubman. High performance scalable image compression with EBCOT. *IEEE Transactions on Image Processing*, 9(7):1158–1170, July 2000. 11, 13, 14, 38, 39, 80

[68] A. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 22(1):1–10, 1976. 21

[69] X. Desurmont, C. Chaudy, A. Bastide, C. Parisot, J.F. Delaigle and B. Macq. Image analysis architectures and techniques for intelligent systems. In *IEE Proceedings on Vision, Image and Signal Processing, Special issue on Intelligent Distributed Surveillance Systems*, volume 152, pages 224–231, 2005. 47