Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2009, Article ID 683820, 18 pages
doi:10.1155/2009/683820

Research Article

Parity Bit Replenishment for JPEG 2000-Based Video Streaming

François-Olivier Devaux and Christophe De Vleeschouwer

Communications and Remote Sensing Laboratory (TELE), Université Catholique de Louvain, 1348 Louvain-la-Neuve, Belgium

Correspondence should be addressed to François-Olivier Devaux, fo.devaux@intopix.com

Received 7 May 2008; Revised 22 October 2008; Accepted 1 June 2009

Recommended by Wen Gao

This paper envisions coding with side information to design a highly scalable video codec. To achieve fine-grained scalability in terms of resolution, quality, and spatial access as well as temporal access to individual frames, the JPEG 2000 coding algorithm has been considered as the reference algorithm to encode INTRA information, and coding with side information has been envisioned to refresh the blocks that change between two consecutive images of a video sequence. One advantage of coding with side information compared to conventional closed-loop hybrid video coding schemes lies in the fact that parity bits are designed to correct stochastic errors and not to encode deterministic prediction errors. This enables the codec to support some desynchronization between the encoder and the decoder, which is particularly helpful to adapt on the fly pre-encoded content to fluctuating network resources and/or user preferences in terms of regions of interest. Regarding the coding scheme itself, to preserve both quality scalability and compliance to the JPEG 2000 wavelet representation, a particular attention has been devoted to the definition of a practical coding framework able to exploit not only the temporal but also spatial correlation among wavelet subbands coefficients, while computing the parity bits on subsets of wavelet bit-planes. Simulations have shown that compared to pure INTRA-based conditional replenishment solutions, the addition of the parity bits option decreases the transmission cost in terms of bandwidth, while preserving access flexibility.

1. Introduction

Our work targets applications requiring a highly scalable access to stored video sequences. Clients logging to the system are expected to have very different profiles, resources, and interests in the content. Common remote interactive actions on the potentially high-resolution video sequences are zooming and cropping on regions of interest (RoI), but also extraction of low-resolution versions of both individual frames and video segments. Hence, scalability in resolution, quality, and spatiotemporal access is required, including an access to individual (fragments of) frames. To achieve such a fine-grained scalability requirement while preserving coding efficiency, we have chosen to encode the sequences with the JPEG 2000 compression algorithm. As temporal redundancy is not exploited due to the INTRA nature of JPEG 2000, the system performances are significantly penalized compared to closed-loop compression schemes such as MPEG, when a segment of consecutive frames has to be accessed. In order to decrease the impact of this drawback, we have adapted conditional replenishment principles to the specificities of JPEG 2000 in a previous work [1]. In the current paper, our main contribution consists in completing the framework proposed in [1] with the introduction of a new replenishment method based on the parity correction of the side information available at the client side. As an interesting feature, it is worth mentioning that the use of parity bits relaxes the usual constraint on encoder-decoder tight synchronization, in the sense that they do not encode deterministic prediction errors but correct stochastic predictive approximations of the image to encode. More specifically, our coding framework is based either on open-loop (INTRA replenishment case) or stochastically—in contrast to deterministically—closed-loop principles. Thereby, it circumvents the drawbacks of conventional closed-loop prediction systems by avoiding systematic temporal propagation of errors. When the reference is not correctly approximated at the client side, the JPEG 2000 INTRA replenishments and the soft decoding of parity bits are expected to increase the stability of the entire system.

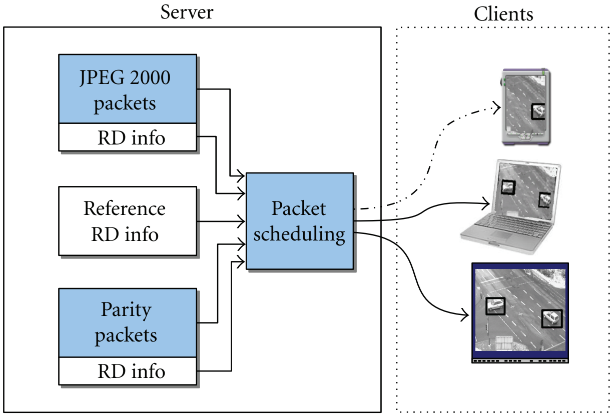

Figure 1: Proposed system architecture. Based on a rate-distortion analysis, the server decides for each precinct either to transmit the corresponding JPEG 2000 packet, to forward a parity packet, or to recommend the use of the available reference to the client. The scheduler takes these decisions at transmission time independently for each user based on its individual requests of interest and available resources.

This is especially relevant when addressing heterogeneous clients dealing with different prediction references or lossy connections, but will not be further considered in this paper. More importantly, we focus our paper on the observation that the addition of the parity replenishment option significantly increases coding efficiency and on the fact that this increased coding efficiency does not alter the flexible and highly scalable nature of the underlying JPEG 2000 codec. That is, the proposed framework is still able to take user interests and channel resources into account at transmission time, independently of the compression stage. This is done by selecting replenishment options on the fly, in a rate-distortion optimized way, given user RoI preferences and instantaneous channel conditions. In final, it also means that the regions of interest can be defined a posteriori, at transmission time, by each individual user.

Figure 1 gives an overview of the proposed system. First, the compressed information composed of JPEG 2000, and parity packets is generated offline and stored on the server, together with information about the rate-distortion tradeoffs associated to the transmission of those packets. Then, at transmission time, the server takes both the user preferences—in terms of RoI—and the channel conditions into account to select the set of rate-distortion optimized replenishment decisions. Formally, the replenishment options consist either in the transmission of (1) a JPEG 2000 packet to refresh the corresponding precinct in the reference available at the client, (2) a parity packet to correct this reference, or (3) a single byte requesting the client to use this reference per se.

The rest of this paper is organized as follows. Section 2 surveys the context and the main characteristics of the proposed framework. Section 3 presents an overview of the proposed replenishment system. Section 4 details the refinement method based on parity coding. Section 5 describes the practical steps leading to the generation of the compressed data and its allocation at transmission time based on the clients requirements. Section 6 presents simulation results and compares the performance of the proposed scheme with MPEG and JPEG 2000 schemes. Section 7 draws the conclusions of the study.

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 prerecorded content based either on hybrid (INTER) or JPEG 2000 (INTRA) compression formats. We present in this section an extension of the browsing scenario described in [1].

In the envisioned scenario, a graphical user interface (GUI) enables the human controller to visualize the chronology of recorded and possibly preanalyzed 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). One 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 1 considers a content captured at 15 fps with a still 2 Mpixels camera and reviews the four access scenarios involved in the browsing session 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 timeline 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 384 × 288 window in a randomly selected frame of the original video sequence.

For each scenario, Table 2 compares the average bitrate required to access a typical surveillance content based on six distinct codecs. We first consider the 4 initial coding schemes (J2K, JPR, and AVC columns) and will comment the last two columns (SVC and AVC FMO) afterwards. For each coding scheme and each spatial resolution, the encoding parameters have been tuned to reach an approximate PSNR of 35 dB, offering a roughly equivalent visual quality for all scenarios. In Table 2, the bandwidth is defined in kbits/sample or kbits/s depending on whether the scenario considers the access to an individual frame or to a video segment. J2K (column 1) encodes and decodes the video images based on the JPEG 2000 algorithm. JPR (JPEG 2000 and Parity Replenishment—column 2) refers to the original solution described and validated in this paper. 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 1. As AVC

Table 1: Content access scenarios definition. Content has been captured at 15 fps, with a 2 Mpixel camera.

	Scenario	Encoded signal resolution	Displayed fraction of initial image
1	Timeline 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 (spatiotemporally) random frame segment	1536 × 1132	1/16

Table 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 JPR methods, a single fine-grained codestream is generated and can address the four envisioned scenarios. SVC and AVC streams are generated to strictly target the four predefined scenarios. As the first and last scenarios are related to the transmission of arbitrary (part of) frames, the bandwidth consumption is measured in kbits/sample. For the other scenarios, the bitrate is expressed in kbits/s.

Scenario	J2K	**JPR**	AVC (I+14P)	AVC (All I)	SVC	AVC FMO (I+14P)
1 (Frames: kbits/sample)	24	**21**	20	20	20	20
2 (Video: kbits/s)	1020	**181**	78	840	93	78
3 (Video: kbits/s)	702	**147**	215	2190	251	101
4 (Frames: kbits/sample)	32	**28**	494	415	537	57

is not supposed to provide spatiotemporal 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. Specifically, 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 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 JPR solutions in scenarios 3 and 4. Interestingly, we observe that our proposed JPR solution preserves the advantages of J2K, while smoothing out its main drawback. Specifically, JPR 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.

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 [2], 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 2. This is illustrated by column 5 (SVC) in Table 2. There we present an SVC solution for which the lowest resolution has been encoded based on a I + 14P GOP structure. For the second and third levels of resolution, frames are predicted both based on the immediately lower resolution and on the previous frame. To improve random access capabilities, the fourth and finest resolution only exploits the lower resolution as a reference (and not the previous frame). We observe in Table 2 that SVC achieves slightly worse performance as the four versions envisioned for AVC I + 14P. This is not surprising since SVC encounters some (minor) penalty when embedding the four AVC versions in a single bitstream. Beyond that example, it is also worth mentioning that the medium-grained scalable (MGS) version of SVC supports rather fine "on the fly" adaptation of entire frame quality, based on a smart design of temporal prediction loops and on the frequency-based partitioning of enhancement coefficients. However, such MGS setting does not support neither RoI-based transmission, nor random temporal or multiresolution access to video. The performances of SVC are compared to our system and to AVC in Section 6.

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. (e.g., [3] 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.) The last column in Table 2 presents the bandwidth requirements corresponding to the four envisioned scenarios when the AVC I+14P codec considers independent slices of 64×64 pixels, thereby significantly improving the bandwidth requirement when random spatial access is required (for scenarios 3 and 4).

Bottom line, we conclude that, for a predefined set of access scenarios characterized by a given set of targeted resolutions or bit budgets, 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

Table 3: Summary of codecs comparison. Fixed scalability refers to the flexibility resulting from the creation of several embedded versions of a compressed content in a single pre-encoded codestream. Adaptive scalability refers to the additional flexibility arising when the transmitted content can be adapted online to user requirements and channel conditions.

Codec	Compression efficiency	Fixed scalability	Adaptive scalability
J2K	Low	High	High
Proposed JPR	**Medium**	**High**	**High**
AVC	High	Low	Low
SVC and/or FMO	Medium-High	High	Low-Medium

slices. However, despite this observation, JPEG 2000-based solutions still remain attractive due to their inherent fine-grained embedded nature, which gives them the ability to deal with heterogeneous bandwidth constraints and RoI user requests. With JPEG 2000 or the proposed replenishment framework, there is no need to work with sophisticated decoder architectures, able to handle a discrete set of (embedded) versions of the same content, encoded with a discrete set of distinct quality and resolution levels. With replenishment-based solutions, the client simply handles and decodes conventional JPEG 2000 and parity 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.

A summary of this comparison is presented in Table 3. There, we differentiate the *fixed scalability* and the *adaptive scalability*. By *adaptive scalability*, we refer to the dynamic adaptation of transmitted content according to the requirements defined by each individual client at transmission time. In contrast, the *fixed scalability* refers to the preparation and exploitation of (embedded) codestreams(s) dedicated to a discrete set of predefined transmission conditions. The ambiguous characterization of the SVC codec in Table 3 reflects the fact that the compression efficiency and adaptive scalability of SVC depend on the envisioned scenarios and parameterization of the codec. More interestingly, based on Table 3, we conclude that J2K and proposed replenishment-based methods outperform other approaches in terms of *adaptive scalability*.

Hence, the core of paper consists in explaining and demonstrating how parity-based 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, thereby reaching the performance presented in the JPR column of Table 2.

3. Conditional Replenishment Mechanisms

Conditional replenishment has been introduced by Mounts [4] in the early years of digital video coding. The basic concept is that only the parts of the current frame that significantly differ from a reference maintained at the receiver are transmitted. Since its introduction in 1969, conditional replenishment has been exploited in several papers, like in [5] where it is used as a simple video coding method for multicast distribution. Our work extends the conditional replenishment principle in three directions. First, the replenishment is achieved in the wavelet domain instead of the pixel domain. Second, besides the possibility to approximate the segment to transmit with the reference or to refresh it with INTRA data, we propose an alternative replenishment approach which consists in correcting the reference with parity bits, as depicted in Figure 2. Third, the RD optimal replenishment decisions are taken in a way that can integrate the definition of user preferences at transmission time. Hence, user actions on the content like zooms and crops can be taken into account at transmission time to decide about packet forwarding decisions.

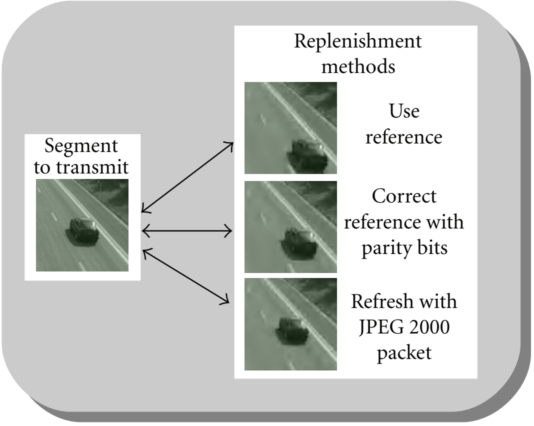
Replenishment methods
Segment to transmit
Use reference
Correct reference with parity bits
Refresh with JPEG 2000 packet

Figure 2: Given an image segment to transmit, the system recommends the client to use one of the 3 available replenishment options, depending on rate-distortion analysis.

The wavelet domain supports the spatial and resolution scalability of our system, in a similar way as for JPEG 2000. A succession of *Discrete Wavelet Transforms* (DWTs) are applied recursively to the original image, as illustrated in Figure 3. Each transform creates four *subbands* (LL, HL, LH, and HH) containing the vertical and horizontal low (L) and high (H) frequencies of the original data. In order to obtain spatial scalability, we divide each resolution in rectangular zones, corresponding to the *precinct* concept defined in the JPEG 2000 standard [6]. Hence, since JPEG 2000 and parity data packets are generated independently for each precinct, individual decisions can be taken for each precinct, offering the spatial and resolution scalability mentioned above.

- In order to reach a fine granularity in quality, the coefficients inside a precinct are described sequentially by bit-planes from most to least significant. In practice, subsets of consecutive bit-planes are encapsulated within distinct packets, thereby defining a hierarchy of layers. This description by layers is implemented both for parity and JPEG 2000 packets

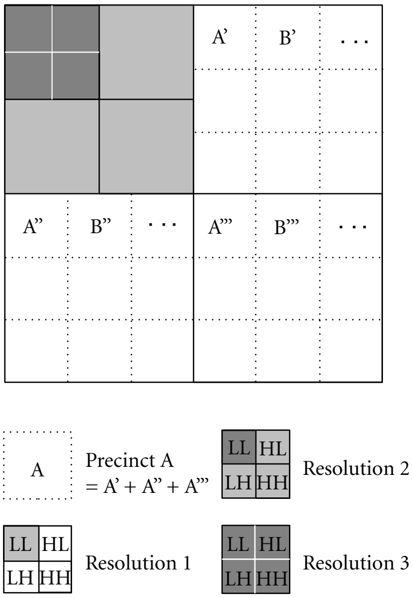

Figure 3: JPEG 2000 wavelet subbands and precincts. A first discrete wavelet transform is applied on the original image, generating four *subbands*. The DWT is then applied recursively on the LL subbands containing the low frequencies of each resolution. A *precinct* is a spatial subdivision of a resolution and corresponds to the same spatial zones in each subband.

> and plays a dual role in adaptive quantization and progressive transmission, by realizing a sequence of successively refined uniform quantizers [7]. Compared to less significant bits, 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.

The proposed coding framework brings interesting advantages when compared to closed-loops frameworks. Although a reference is also exploited in our work, the transmitted data are either INTRA JPEG 2000 packets, or parity bits which do not encode deterministic prediction errors but rather correct stochastic prediction errors, thereby tolerating a certain desynchronization between the encoder and the decoder. This is particularly important for transmissions in noisy environments as well as when serving heterogeneous clients with different prediction references. To address the latter case, our system considers multiple protection/redundancy levels for each parity packet. Formally, it computes several sequences of parity bits for each subset of bit-plane defining a precinct layer, each sequence corresponding to a distinct compression ratio and correction capability. At transmission time, the amount of parity bits to convey for a given precinct layer can thus be adjusted for each client based on the quality of the reference, thereby relaxing the closed-loop constraint associated to conventional predictive coding schemes [1].

4. Parity Refinement

This section describes the way our system generates parity bits to correct the side information provided by the reference stored at the client side.

We first present the principles of video coding using side information. Then, we explain how we have exploited the temporal correlation between consecutive frames to initialize the iterative decoding, as well as the spatial correlation inherent to an image source. Finally, we detail the practical implementation of the parity refinement module.

4.1. Video Coding with Side Information: Introduction. Video Coding with side information is based on the Slepian-Wolf theorem [8] published in 1973. Let us consider two statistically dependent signals X and 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* with rate R_X and R_Y, as long as the two coded streams are decoded *jointly* and the following conditions are met [8]:

$$\begin{aligned} R_X &\geq H(X \mid Y), \\ R_Y &\geq H(Y \mid X), \\ R_X + R_Y &\geq H(X, Y). \end{aligned} \tag{1}$$

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 $H(X \mid Y)$ 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 *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 at the client of the side information Y.

We learn from the analysis of the conditional entropy $H(X \mid Y) = H(X) - I(X; Y)$ that two factors are important to decrease the rate at which we can transmit X.

(i) *$I(X; Y)$, 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 and will be studied in Section 4.3.

(ii) *$H(X)$, 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 $H(X)$. (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.) This will be studied in Section 4.4.

Since the proofs of the Slepian-Wolf and Wyner-Ziv theorems are asymptotical and nonconstructive, different ways of implementing such coding systems have been proposed [9–11]. In particular, several distributed video coding systems have exploited the wavelet transform [12–15]. Similar to those previous frameworks, our work considers the side information Y as a noisy version of X, and conventional channel coding techniques are used to correct the side information. At the encoder, *parity bits* typically generated with Turbo Codes and LDPC are calculated for X and are transmitted to the decoder where they are used to correct Y. The correlation between X and Y is often associated to a *Virtual Channel*, as the parity bits aim at correcting the errors introduced by this channel. This process is depicted in Figure 4.

While video coding with side information has been studied extensively for enabling low-complexity video encoding, a significant interest to use these techniques for *flexible decoding* has been observed in the last years [16]. Our work also targets such flexible decoding, in which encoders generate a single compressed bit stream that can be decoded in several different ways, depending on each user requirements and transmission conditions.

We now highlight the specificities of our proposed system.

4.2. Video Coding with Side Information: Specificities of the Proposed System. Since a major motivation of this paper is scalability, parity replenishment must preserve the granularity of the compressed data in terms of spatial access, resolution, and quality. Hence, parity packets are generated independently for each precinct and encoded in several embedded quality layers, each layer gathering a certain number of consecutive bit-planes. The optimal calculation of the truncation points, that is, the optimal number of bit-planes for these quality layers, should be guided based on a careful rate-distortion analysis, as done for JPEG 2000 [6]. In practice, for simplicity, the same truncation points optimally calculated for the corresponding JPEG 2000 packets have been used in our study. (Although the distortion of each parity and JPEG 2000 layer is the same, the number of bits used to generate the compressed data is not the same for both methods. Hence, the chosen truncation points are likely to be globally (at image level) suboptimal for the parity replenishment. However this suboptimality is partly compensated by the fact that the allocation of packets to the image precincts is guided by a globally optimal RD convex-hull analysis (see Section 5.1).)

Of all practical error correction methods known to date, LDPC [17] and Turbo codes [18] come closest to 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 $M \times K$. At the encoder side, the sequence of input bits belonging to the hierarchy of layers of 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. The initial probability distribution of the ith 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 the sequence of input bits $\mathbf{X}$.

Our decoding model is illustrated in Figure 5 as a *factor graph* [19]. A factor graph is a bipartite graph that expresses which variables (circles) are arguments of which local functions (squares). The local functions f_i represent the linear transformation $\mathbf{H}$, and the results of that transformation are the parity bits 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 the spatial correlation between precinct coefficients into account during the decoding process.

Decoding is achieved using the sum-product algorithm [19]. This algorithm aims at computing the marginal posterior probabilities $P(Y_i = 1 \mid \mathbf{Z}, \mathbf{H})$ for each i, based on an iterative process consisting of message exchange and probability distribution updates.

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. Typical models used to describe such correlation follow the Laplacian or Gaussian distributions. In this work, we have adopted the second one. Formally, let $C^{n,r,t}$ denote the random variable associated to the nth wavelet coefficient of the rth resolution at time t. We then 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

$$P\left(C^{n,r,t+1} = m \mid C^{n,r,t} = n\right) = \frac{1}{\sqrt{2\pi\sigma^2_{r,t}}}\, e^{-(n-m)^2/2\sigma^2_{r,t}}. \quad (2)$$

Based on the coefficient distribution, we can compute the distribution for $B_k^{n,r,t+1}$, the random variable associated to the kth most significant bit of $C^{n,r,t+1}$. For this purpose, we introduce $\beta_k(m)$ to denote the function extracting the value

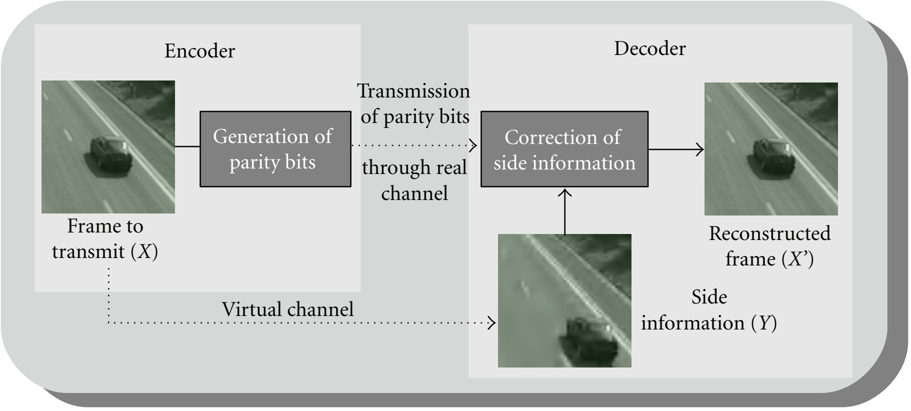
Encoder
Frame to transmit (X)
Generation of parity bits
Transmission of parity bits
through real channel
Virtual channel
Decoder
Correction of side information
Reconstructed frame (X')
Side information (Y)

FIGURE 4: 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.

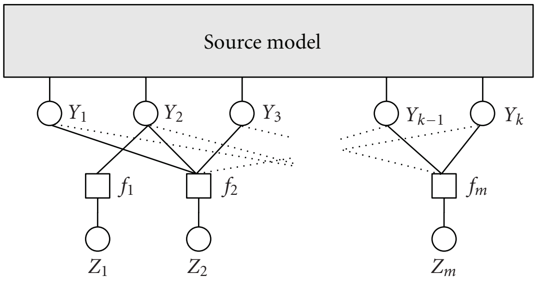

FIGURE 5: Factor graph representing the source model and the LDPC code. At each iteration of the decoding process, messages describing local distributions are exchanged between nodes inside the LDPC code and with the source model.

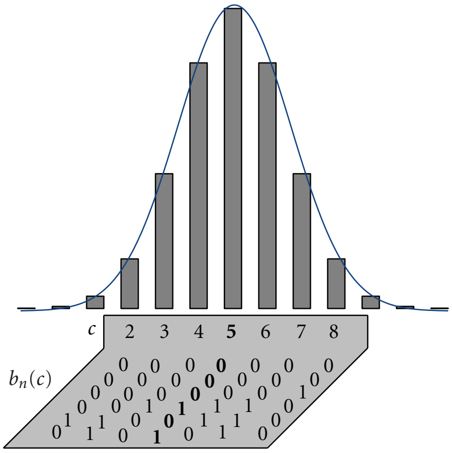

FIGURE 6: Probability distributions of coefficients have to be adapted due to the representation in bit-planes. Bits representing the coefficient are weighted by a Gaussian distribution centered on the realization of the coefficient in the previous frame.

of the kth bit of coefficient m. Hence, we have $B_k^{n,r,t+1} = \beta_k(C^{n,r,t+1})$ and

$$P\left(B_k^{n,r,t+1} = 1 \mid C^{n,r,t} = n\right) = \frac{1}{\sqrt{2\pi\sigma_{r,t}^2}} \sum_{m=-\infty}^{\infty} e^{-(n-m)^2/2\sigma_{r,t}^2} \beta_k(m). \tag{3}$$

Figure 6 illustrates this Gaussian weighting of the coefficients bits. In the chosen example, nonsignificant 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 notation simplicity, we now omit the time index t ($C^{n,r,t}$ is now denoted $C^{n,r}$) and denote the variance parameter for resolution r by σ_r^2. This value is computed for each frame at the encoder and transmitted to the decoder.

This frame level approximation can be spatially refined by taking into account the spatial fluctuations of the variance. Indeed, important and local modifications of the content between consecutive frames are reflected throughout several resolutions. This is illustrated in Figure 7 which presents the spatial maps of absolute prediction errors for distinct resolutions and a pair of frames of the *Speedway* sequence. (Details regarding the *Speedway* sequence are provided in Section 6.) 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 less significant and more localized modifications of content 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 index and by estimating this variance at a given resolution 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

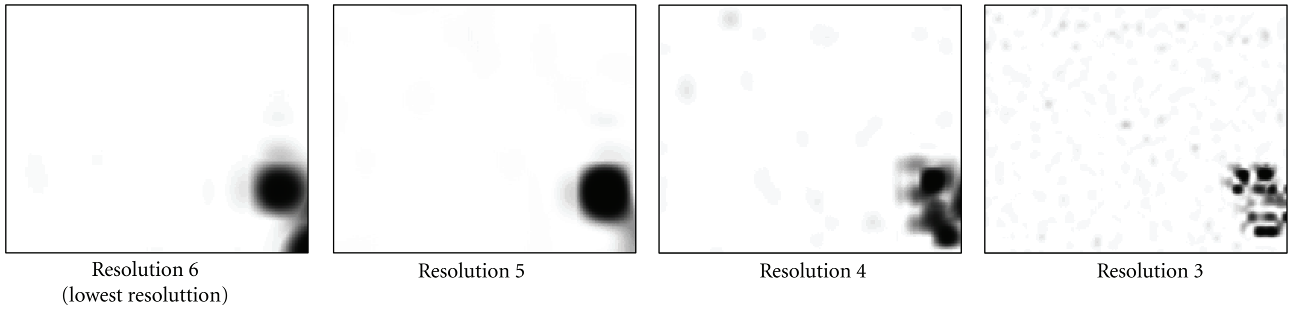

Figure 7: Maps of absolute difference between precincts in two consecutive frames, by decreasing order of resolutions starting from the lowest resolution on the left, for first two frames of the *Speedway* Sequence [20]. In this figure, resolutions have been resampled to the same size, and the absolute difference values of each resolution have been rescaled between 0 (white regions) and 255 (black regions).

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. (Note that the resolution index r increases when the resolution decreases, thereby following the JPEG 2000 terminology.) 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 Euclidean distance between coefficient $C^{m,r}$ and coefficient $C^{n,r}$, the local variance $L_\sigma^{n,r}$ is defined as

$$L_\sigma^{n,r} = \left(\sum_{m=-\infty}^{\infty} \frac{1}{d_{m,n,r}}\right)^{-1} * \sum_{m=-\infty}^{\infty} \frac{E_{sq}^{m,r}}{d_{m,n,r}}. \tag{4}$$

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

$$\sigma_{n,r}^2 = \sigma_r^2 \frac{L_\sigma^{n,r+1}}{\sigma_{r+1}^2} \tag{5}$$

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 are discussed in Section 6.

4.4. Exploiting Spatial Correlation of Precinct Coefficients. 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.

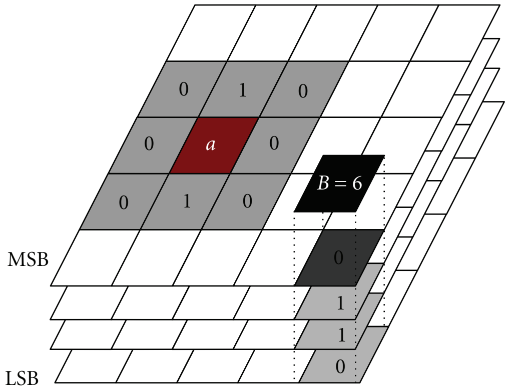

Figure 8: *Bit-plane representation*. The coefficient *B* which is equal to 6 is represented in binary form (0110) through the four bit-planes. Its MSB is insignificant, and the remaining bits of the coefficient are significant. *Bit-plane context modeling*. The context of bit *a* is calculated based on its eight neighbors significance as well as its own significance state.

Spatially, the value of a bit inside a wavelet bit-plane can be estimated mainly on the value of its neighbors as well as the value of the coefficient in more significant bit-planes [21], which is related to the concept of *significance*. 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. A precinct and its representation in bit-planes is illustrated in Figure 8.

In our work, the *image modeler* is based on the Embedded Block Coding with Optimized Truncation (EBCOT) algorithm [22], used in the JPEG 2000 standard [6]. According to this algorithm, a bit is classified in one of the 19 different categories called *contexts*, based on the significance of its eight contiguous neighbors and its own significance state. The statistics of the bits can differ highly, depending on their context. 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

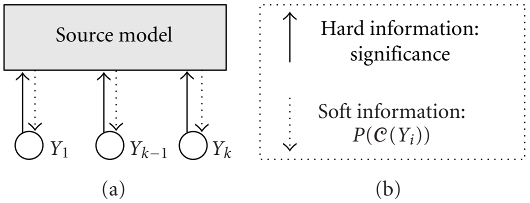
Source model
Y_1
Y_{k-1}
Y_k
Hard information: significance
Soft information: $P(\mathcal{C}(Y_i))$
(a)
(b)

Figure 9: Hard decisions are transmitted from each LDPC node, and soft information calculated in the Source Modeler is sent back.

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

In practice the bit probability distributions associated to each context are approximated by frequencies of occurrence of bit values. Those occurrence frequencies can be calculated in different ways. They could for example be estimated on the image at hand and transmitted to the decoder as a side information. A second option could consist in approximating frequencies of occurrence based on previously transmitted frames. This approach has the advantage to adapt the distributions to the video at hand, without requiring explicit transmission of the distributions. In our simulations, to limit computations, we have however decided to rely on a hard-coded distribution, computed based on a large 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 $c = \mathcal{C}(i)$ corresponding to the hard decisions taken for the neighbors of Y_i and returns the soft information $P(Y_i = 1 \mid \mathcal{C}(i) = c)$, as illustrated in Figure 9. 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.

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 in presence of spatial correlation information.

4.5.2. LDPC Codes. In our system, a discrete set of distinct LDPC matrices **H** have been generated. As explained above, for each precinct layer to encode (The layers can be embedded or not, depending on how the application wants to tradeoff compression efficiency and storage resources. The embedded case encodes a hierarchy of layers, for which each parity layer only encodes the bit-planes to add to previous layers to increment the quality. In contrast, the nonembedded solution encodes the entire set of bitplanes corresponding to a level of quality. This second aproach achieves higher coding efficiency, mainly because LDPC codes are more efficient when handling long codewords. However, it also results in storage overhead, due to the inherent redundancy between quality packets.), multiple parity packets are generated, each packet corresponding to a distinct matrix **H**, that is, to a distinct compression ratio and a distinct correction capability. (The generation and storage of these multiple **H** matrices could be avoided by using fountain codes [23], like the LT [24] and Raptor codes [25], in place of LDPC codes.) At transmission time, the smallest of these matrices that are able to correct the reference precinct is selected, and the generated parity bits transmitted. In Section 5.1, we describe how this choice can be adapted to individual clients at low computational cost.

Regarding the practical implementation of the codes, we have considered regular LDPC codes with a standard bipartite graph structure [26]: 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 [27]. 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 [28, 29].

4.5.3. Interleaving of Bits within a Precinct. Since the LDPC codeword length is limited to 8000 bits, large precinct layers result in several codewords. To limit the transmission overhead associated to the definition of the parity compression ratio, all codewords associated to a precinct are encoded based on the same parity length M. However, the bit-planes composing the precinct layer do not experience the same BER. (The Bit Error Ratio (BER) considered here is related to the virtual channel between X and Y in Figure 4, that we assume binary and symmetric.) Typically, LSBs usually have a higher BER than MSBs. For this reason, bits of distinct bit-planes of a precinct are interleaved before parity bits

Table 4: Replenishment mechanisms selected for each precinct at various rates for the first replenished frame of the *Speedway* sequence.

Precinct index	Rate 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 5: Summary of exploited correlation.

Temporal correlation	⇒*Gaussian distribution of coefficients*
Coherence across resolutions	⇒*Gaussian variance spatial fluctuation estimation*
Spatial correlation	⇒*EBCOT context modeling*

computation, to make the BER of all codewords more or less uniform. The interleaving process is illustrated in Figures 10 and 11.

4.5.4. Contexts. Practically, only a subset of the 19 contexts defined by the EBCOT [6] 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.

Table 5 summarizes how our system exploits the correlation to improve its coding efficiency.

5. Generation and Allocation of Compressed Data

This section first explains how the *Rate-Distortion* (RD) optimal selection of available replenishment options is performed for each precinct, according to instantaneous channel conditions and user preferences. It then describes how the resources associated to the precomputation and storage of the RD information required to guide the RD optimal selection can be drastically reduced based on approximations of the temporal evolution of the prediction errors and distortions. Eventually, the section summarizes how pre-encoded data and associated rate-distortion information is generated to be stored on the server. For each precinct, several—INTRA and parity layers—replenishment options are considered, and the corresponding bitstream segments and RD information components are precomputed.

5.1. Rate-Distortion Representation and Optimal Allocation. During the streaming, the server tries to adapt its scheduling decisions to the needs and resources of the client. In practice, this is done by selecting the replenishment options that maximize the reconstructed image quality under a given bit budget constraint and an estimated reference frame. In our context, the replenishment options correspond either to the use of the available reference or to the transmission of a specific JPEG 2000 or parity layer, each option resulting in a particular rate-distortion tradeoff for the envisioned precinct.

The problem of rate-distortion optimal allocation of a bit budget across a set of image blocks characterized by a discrete set of RD tradeoffs has been extensively studied in literature [30–32]. Under strict bit budget constraints, the problem is hard and relies on heuristic methods or dynamic programming approaches to be solved [32]. In contrast, when some relaxation of the rate constraint is allowed, Lagrangian optimization and convex-hull approximation

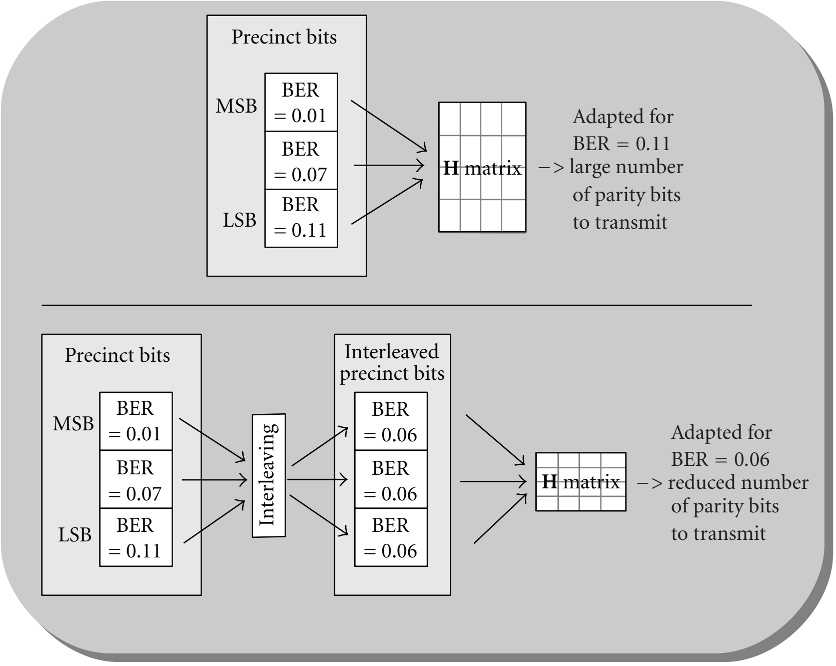
Precinct bits
MSB
BER = 0.01
BER = 0.07
LSB
BER = 0.11
H matrix
Adapted for BER = 0.11 −> large number of parity bits to transmit
Precinct bits
MSB
BER = 0.01
BER = 0.07
LSB
BER = 0.11
Interleaving
Interleaved precinct bits
BER = 0.06
BER = 0.06
BER = 0.06
H matrix
Adapted for BER = 0.06 −> reduced number of parity bits to transmit

Figure 10: *Precinct layer bit-planes interleaving.* When multiple codewords are generated for a given precinct, interleaving of the bits enables to make the distribution of errors uniform and consequently reduces the size of the matrix **H** required to correct the precinct, reducing the number of parity bits to transmit.

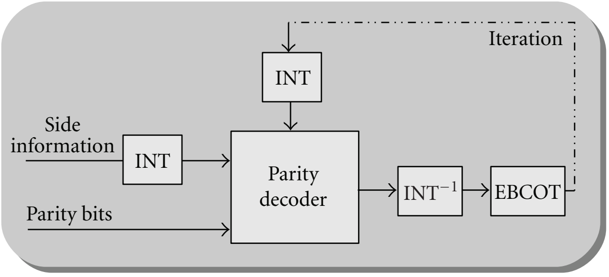

Figure 11: *Interleaving operations at decoding.* To decode a precinct, an interleaved (INT) version of the side information and the corresponding received parity bits are fed to the parity decoder. At each iteration of the sum-product algorithm, the hard decisions of each bit are deinterleaved (INT^{-1}) and transmitted to the EBCOT. The EBCOT output, consisting in soft information for each bit, is interleaved and fed into the parity decoder for the next iteration.

can be considered to split the global optimization problem in a set of simple block-based local decision problems [30, 31]. The convex-hull approximation consists in restricting the eligible transmission options for each block 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 precinct convex-hulls, in decreasing order of distortion reduction per unit of rate.

This optimal RD allocation process is illustrated in Figure 12, where the possible replenishment decisions are arranged in a rate-distortion graph.

In a classic application scenario, the server adaptively selects the packets to convey to the client as a function of the instantaneous network conditions and user interest, which might weight the distortion measures in Figure 12 [33]. As a consequence, the reference image available at the client directly depends on previous replenishment decisions. In particular, the reference of a given precinct depends on the moment and quality level at which the precinct has last been refreshed, either based on JPEG 2000 or parity data. This is the source of tremendous computational and storage effort when precomputing the possible RD tradeoffs for a given frame. Specifically, both the reduction of distortion associated to a JPEG 2000 or parity replenishment and the parity rate required for a reference correction have to be computed for any possible reference. In the next section, we explain how this rate and distortion information can be approximated based on the quality improvement and parity length computed between pairs of consecutive frames, thereby reducing both computational and storage resources at the server.

5.2. Practical Implementation: Storage and Computational Resources Savings. We first consider the approximation of the quality improvement associated to a refreshment. In [33], we have proposed to estimate the squared error (SE) distortion between distant precinct based on a weighted sum of the distortion between consecutive intermediary precincts. The approach has been presented and justified in detail in [34]. We now summarize the main lessons from this study.

As a first step, in [34], we have analyzed how the SE associated to the approximation of a precinct by its last

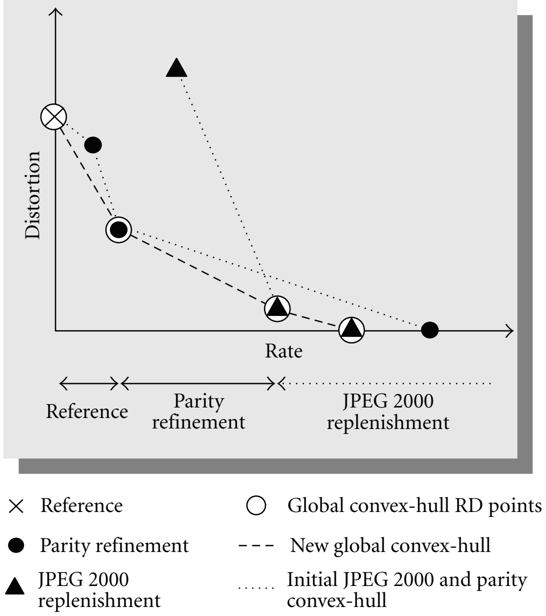

Figure 12: Rate-distortion representation of the replenishment decisions for a given precinct. Depending on the available bit-rate, the client will use the reference (cross) and receive a parity packet (dot) or JPEG 2000 packets (triangles). The original JPEG 2000 RD points and the resulting replenishment decisions lie on convex-hulls.

reference is affected by the time elapsed between the last refreshment and the current time. We have observed that the SE increases with the temporal distance much more significantly for the low frequency resolutions than for high-frequency resolutions. A second observation is related to the fact that the SE increases consistently during several frames at low-resolutions, while it rapidly saturates for higher frequencies. This is in accordance with the fact that the temporal correlation is mostly present in the low frequencies. Our proposed SE approximation integrates these two observations. It is motivated by Figure 13, which depicts the hierarchy of layers associated to frames between time $t-k$ and t. In this figure, we denote $d^{k,\kappa}(i,t)$ to be the distortion measured when approximating the ith precinct of frame t, based on the κ first layers of the corresponding precinct in frame $(t-k)$. We also denote $\kappa_{\max}$ to be the highest quality level. 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 $d^{0,q}(i,t-k)$ 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 the frame distance.

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

$$\widehat{d^{k,\kappa}}(i,t)=d^{0,\kappa}(i,t-k)+\sum_{l=0}^{k-1}\omega(l,r)d^{1,\kappa_{\max}}(i,t-l). \quad (6)$$

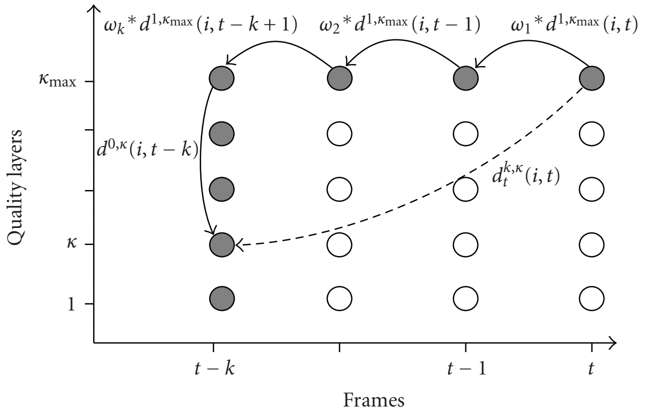
$\omega_k * d^{1,\kappa_{\max}}(i,t-k+1)$
$\omega_2 * d^{1,\kappa_{\max}}(i,t-1)$
$\omega_1 * d^{1,\kappa_{\max}}(i,t)$
$\kappa_{\max}$
$d^{0,\kappa}(i,t-k)$
$d_t^{k,\kappa}(i,t)$
Quality layers
κ
1
$t-k$
$t-1$
t
Frames

Figure 13: 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.

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 the pair of above observations. It appeared from our investigations that the following definition achieves good prediction performances:

$$\omega(l,r)=\alpha(r)e^{-(R-r)}e^{-l}, \quad (7)$$

where R corresponds to the total number of resolutions, r corresponds to the precinct resolution index and is numbered as previously in this work ($r=R$ for the lowest resolution), and $\alpha(r)$ has been defined in our simulations as

$$\alpha(r)=\frac{30}{1+(R-r)}. \quad (8)$$

In the second part of this section, we explain how the rate associated to parity refreshments can be adapted to the reference available at the client. The problem lies in the fact that the number of parity bits required to correct a reference depends on the virtual channel noise, which is specific to each client session since the reference to correct depends on previous replenishment decisions. Hence, precomputation 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. Again, an extensive description is provided in [34], and we only summarize our main findings in the sequel.

Formally, let $s_P^{k,\kappa}(i,t,q)$ denote the number of parity bits required to correct the first q layers of precinct i at time t based on the first κ layers of the corresponding precinct at time $(t-k)$. As a main outcome of the study carried out in [34], we have found that the increment of parity rate associated to a degradation of the virtual channel can be approximated as a linear function of the virtual channel bit error rate (BER) increment. Formally,

$$\begin{aligned}\widehat{s_P^{k,\kappa}}(i,t,q)&=s_P^{1,q}(i,t,q)+\mu(r)\\&\quad*\left(\mathrm{BER}^{k,\kappa}(i,t,q)-\mathrm{BER}^{1,q}(i,t,q)\right)+\nu(r),\end{aligned} \quad (9)$$

where $\mu(r)$ and $\nu(r)$ are estimated for each resolution and denote the parameters of the linear approximation. In the above equation, $\mathrm{BER}^{k,\kappa}(i,t,q)$ denotes the BER measured when approximating the qth layer of the ith precinct of frame t, based on the κ first parity layers of the corresponding precinct in frame $(t-k)$. A similar investigation than the one followed to approximate the SE values reveals that those BER values can be reasonably approximated based on a weighted sum of the BER computed between pairs of precincts belonging to consecutive frames, the layer of the reference precinct being potentially different from the layer of the second one. The approach is motivated by a decomposition of the path between (frame t, layer q) and (frame $(t-k)$, layer κ) that is similar to the one presented in Figure 13. Formally, we have

$$\widehat{\mathrm{BER}}^{k,\kappa}(i,t,q) = \mathrm{BER}^{1,\kappa}(i,t,q) + \sum_{l=1}^{k-1} \omega'(l,r)\mathrm{BER}^{1,\kappa}(i,t-l,\kappa), \tag{10}$$

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

$$\omega'(l,r) = e^{-\beta_1 l} e^{-\beta_2(R-r)}, \tag{11}$$

where $\beta_1 = 0.01$, $\beta_2 = 0.5$ in our simulations. R corresponds to the total number of resolutions and r to the precinct resolution index.

To complete this section, we refer to [34], which has analyzed the quality of the proposed SE and BER approximations. In particular, it has been shown that the image PSNR penalty induced by those approximations when allocating a given bit budget among the possible precinct replenishment options remains below 0.3 dB. Indirectly, the small value of this penalty reveals that the decoder can tolerate the desynchronization induced by the approximations of the SE and virtual channel BER, thereby demonstrating that our system relaxes the tight synchronization constraint inherent to conventional closed-loop systems.

5.3. Generation of Compressed Data. Based on the above considerations, only the distortion and parity length between two consecutive frames need to be computed and stored on the server as RD information.

Figure 14 illustrates the main steps leading to the generation of the RD information and associated compressed data at the server. The first operation is the discrete wavelet transform. A delay module gives access to the previous frame. The *Squared Error* (SE) distortion between this previous frame, considered as a reference, and the frame to transmit is calculated. The next step is the generation of parity and JPEG 2000 packets for each layer of each precinct. Those packets are then stored with their associated rate-distortion information. For both the parity and JPEG 2000 encoding modes, several quality versions (or layers) of the precincts are thus encoded, each version corresponding to a particular quantization level of the precinct coefficients. In addition, as explained in Section 3, parity replenishment considers multiple packets with distinct correction capabilities for each precinct layer. Note that, for completeness, an optional motion compensation (MC) module is included in Figure 14 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 at Section 6.4. It opens important fundamental questions that have not been investigated in this paper. Hence, the proposed experiments focus on video-surveillance content with a still background and leave the study of a motion compensation engine dedicated to our replenishment framework for future research.

6. Results

In this section, we first present the global performances of the replenishment system integrating parity refreshments. Then, we refine the analysis by quantifying the benefits resulting from the exploitation of the temporal and spatial correlations. Finally, we discuss how those results might impact the design of a motion compensation module.

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

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 10 seconds and is available on the WCAM European project website [20]. The second one is *Caviar*, a video-surveillance sequence presenting people walking in front of a shop. Its frame rate, resolution, and length are similar to that of *Speedway*. It is available on the CAVIAR project website [35].

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 128 × 128, and code-blocks have a size of 64 × 64.

Figures 15 and 16 compare for both sequences the performance of our system with conventional (Motion) JPEG 2000, MPEG4-AVC, and SVC solutions. The performances of the proposed system, the JPEG 2000, and SVC solutions refer to the transmission of a single pre-encoded content at multiple rates. In contrast, AVC relies on multiple bitstreams, one for each targeted rate. Three replenishment mechanisms are considered: *JPEG 2000* Replenishment (JR), *Parity* Replenishment (PR), and *JPEG 2000 and Parity* Replenishment (JPR). 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. The SVC solution is characterized by a GOP of 16 frames with medium-grain SNR scalable layers (MGS).

We observe in Figure 15 for the *Speedway* sequence that, unsurprisingly, the standard JPEG 2000 algorithm appears to

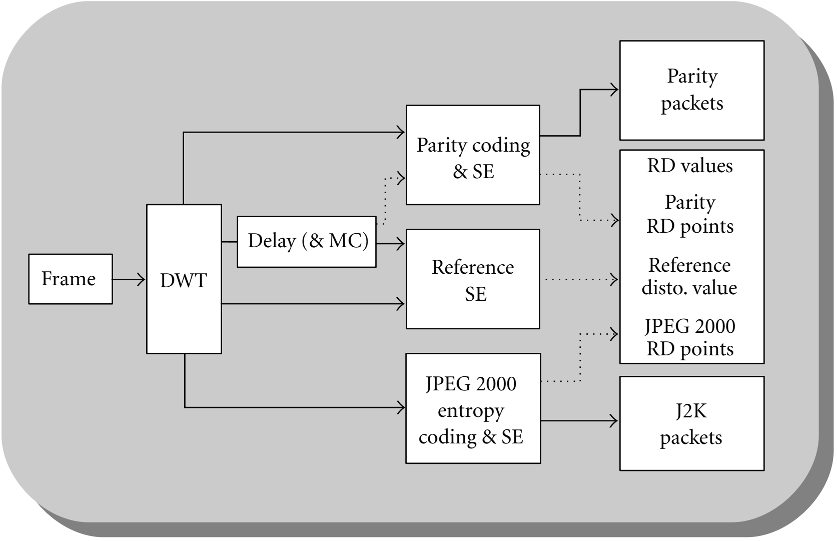

Figure 14: Generation of compressed data and rate-distortion information at the encoder side. The main modules consist in the discrete wavelet transform, the delay (and optional motion compensation) module, the generation of Parity, and JPEG 2000 data with its associated squared error (SE) and the calculation of the reference squared error.

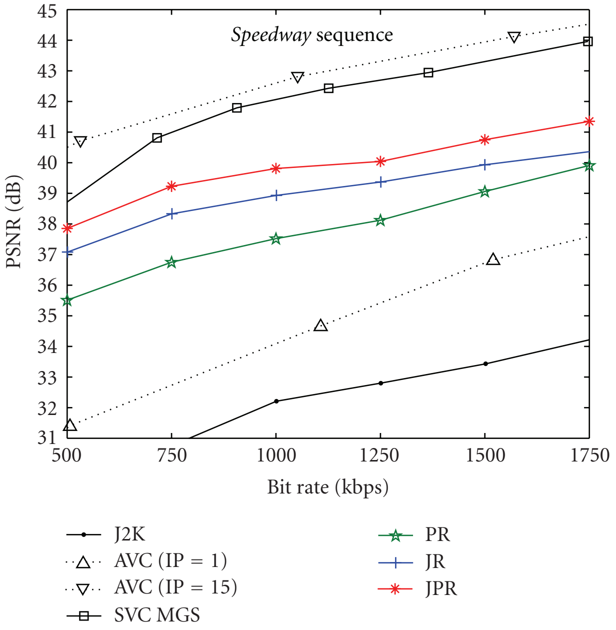

Figure 15: Performances of the proposed system with different combination of Parity and JPEG 2000 replenishments (JR, PR, and JPR), MPEG4-(AVC), and the purely INTRA JPEG 2000 coding scheme (J2K) for the *Speedway* sequence. Frame rates and encoding parameters are defined in the text.

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 and SVC, which do not offer an independent access to each frame nor the possibility to define RoI at transmission time, the replenishment results are convincing, given the increased flexibility offered by these methods and their capability for efficient integration in a low-complexity server (see [1] and Section 5.2). At 500 kbps, JPR is 6.5 dB above AVC IP-1, 1 dB below SVC, and 3 dB below AVC IP-15.

Figure 16 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 is less noisy than *Speedway*, and the temporal correlation is higher, favoring parity coding as explained in Section 4.3.

A deeper analysis of the chosen replenishment options is provided in Table 4. 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. 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 refreshes are more efficient than JPEG 2000. This relative efficiency decreases with the increasing resolutions. At finer resolutions, JPEG 2000 replenishments are chosen because of the efficiency of the entropy coding engine, especially when dealing with long runs of zero coefficients.

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

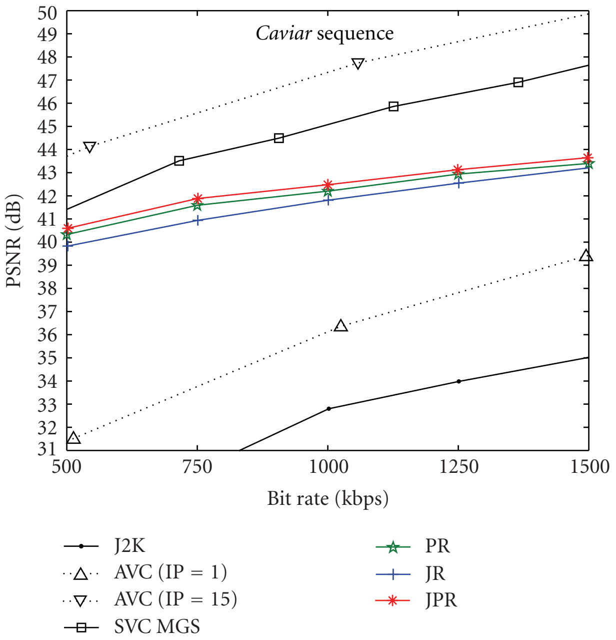

Figure 16: Performances of the proposed system with different combination of Parity and JPEG 2000 replenishments (JR, PR, and JPR), MPEG4-(AVC), and the purely INTRA JPEG 2000 coding scheme (J2K) for the *Caviar* sequence.

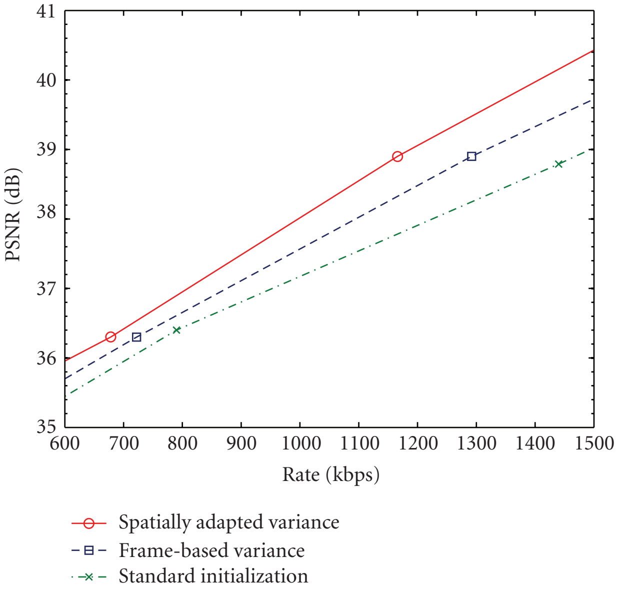

Figure 17: Comparison of the three methods exploiting the temporal correlation between frames in the wavelet domain.

option, passes by a JPEG 2000 refresh, and finally goes through two parity refreshes of increasing quality at high bitrates. This is reflected in Table 4 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 these two replenishment options (between 1000 kbps and 3000 kbps).

Quite surprisingly, in precinct number 2, the JPEG 2000 refreshes are more efficient than the parity refreshes 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.

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 temporal evolution of prediction errors is spatially coherent across resolutions. To integrate this observation into the Gaussian model, a spatial refinement of the Gaussian distribution variance has been adopted.

Figure 17 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). The curve labeled "*Standard initialization*" corresponds to the initialization usually encountered in video coding systems with side information. Specifically, 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 "*Frame-based Variance*" corresponds to the integration of the Gaussian distribution. For the third curve, labeled "*Spatially-adapted Variance*", the system adapts the Gaussian variance based on the prediction error of the corresponding coefficients reconstructed in the lower resolution. The benefit resulting from the improved virtual channel estimation, obtained from refined approximations of the coefficients variance, is obvious. At 1500 kbps, the exploitation of temporal correlation brings a gain of 1.5 dB.

6.3. Spatial Modeling with EBCOT. As presented in Section 4.4, spatial correlation is exploited based on 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 a natural image. (This might for example occur when the prediction results from motion compensation.) In this

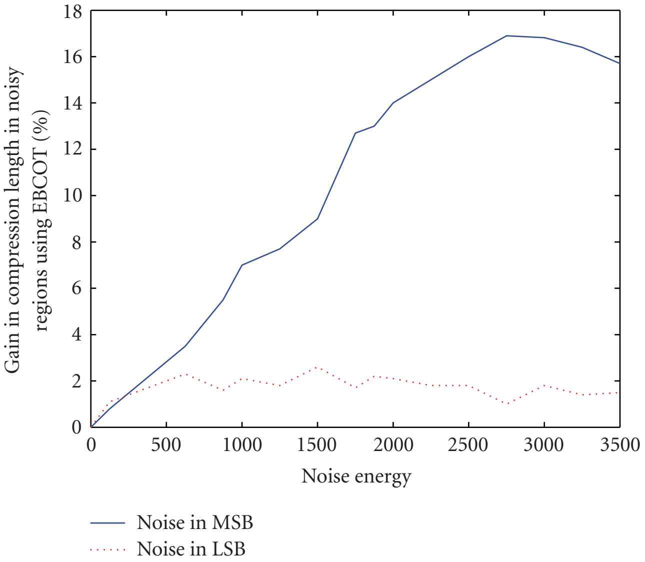

Figure 18: Spatial modeling increases the system performances when the reference available at the client is erroneous. This figure illustrates the increase in compression efficiency brought by the spatial modeler when noise is added to the reference LSB and MSB bit-planes. These results have been generated using code-blocks of the third resolution of the *Speedway* sequence.

case, the EBCOT detects these incoherencies and helps the LDPC decoder by correcting the bits probabilities that do not respect natural image statistics.

To better understand the role of the EBCOT, we have introduced random errors on the reference code-block coefficients, focusing either on LSB or on MSB bit-planes, and have analyzed how the length of parity codes increases with error rate, with and without EBCOT. (Practically, code-blocks bit-planes have been divided in two groups of equal size: LSB bit-planes and MSB bit-planes.) The results of these simulations are presented in Figure 18. 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 bit 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.

A deeper analysis of Figure 18 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.

Note that the above result is surprising. Spatial modeling has been used successfully in other works dealing with distributed [36] or parity [37] coding frameworks, and such conclusions have never been drawn. We now explain the main differences between these earlier works and our approach. The EBCOT has for example been integrated in a joint source-channel image coding system [37]. 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 [37], 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. Moreover, in [37], a parity packet aims at correcting entire bit-planes. Hence, before decoding a given bit-plane, all the previous bit-planes have been 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 drawn 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. In the next section, we consider how the motion estimation metrics can be chosen to create errors which are efficiently corrected by the spatial modeler.

6.4. Note about 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 algorithm 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 options that correspond to localized errors of important magnitude.

Two common metrics used in image and video coding are the MSE (Mean Squared Error) and the SAD (Sum of Absolute Differences). As we are looking for a metric favoring errors with a high magnitude, we propose to use the SRAD (Sum of Rooted Absolute Difference) for this task. Compared to SAD and MSE, the SRAD typically increases (decreases) the impact associated to small (large) errors. The design of a motion compensation that would exploit the specificities of our proposed system has not been considered in this paper but is certainly an interesting research perspective.

7. Conclusions

This paper is an attempt to build a video codec on 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. Such coding paradigm has the advantage of relaxing the constraints inherent to conventional closed-loop codecs. Thereby, it allows for the design of a codec that offers fine-grained scalability in terms of resolution, quality, and spatial access, as well as temporal access to individual frames. Such a solution is also expected to be more robust than a closed-loop video coder to a mismatch between the references available at the encoder and the decoder since parity bits are designed to correct stochastic errors and not to encode deterministic prediction errors. In particular, our work has proved that the decoder can tolerate the desynchronization induced by the approximations of the virtual channel required to reduce computational and storage resources when adapting the forwarded content to user needs and resources. The behavior of our system in error-prone environments has not been studied, but definitely worth a thorough future investigation.

Besides, our work also puts parity-based coding in competition with INTRA JPEG 2000 coding and reference-based replenishment, thereby helping to identify the cases for which parity correction is useful and beneficial. To preserve compatibility with JPEG 2000 INTRA coding, the side information had 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 (1) to exploit the temporal but also spatial correlation among wavelet subbands coefficients and (2) to define the parity bits on subsets of wavelet bit-planes to preserve quality scalability. With that respect, our work has brought three original contributions, namely, (1) the temporal prediction of individual bits of wavelet coefficients through a Gaussian coefficient distribution formalism, (2) the spatial adaptation of the Gaussian variance based on the correlation inherent to adjacent resolutions, and (3) the exploitation of spatial correlation through context-based bit-plane prediction and iterative decoding strategies. Those three mechanisms contribute to improve the decoding capabilities of parity bits.

To evaluate the performance of the proposed system, we have compared conventional AVC and SVC codecs to two instances of our proposed rate-distortion optimized replenishment framework. Both instances rely on preencoded content and can adapt their RD optimal scheduling decisions to user requirements and preferences in real time. In the first instance, the replenishment of the reference image is restricted to JPEG 2000 packets. In contrast, the second instance selects the RD optimal refreshment options within a precomputed set of JPEG 2000 and parity packets. While JPEG 2000 packets induce a complete INTRA refreshment of the reference, parity packets refine this reference using channel coding techniques and LDPC codes. Simulations with video-surveillance sequences have shown that the addition of parity bits offers significant improvement compared to pure INTRA refresh, without reaching the coding efficiency of motion compensated algorithms. Hence, it 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.

Acknowledgments

This publication has been partly funded by DGTRE project WALCOMO, Grant no. 616448. The authors are funded by the Belgian NSF.

References

[1] F.-O. Devaux, C. De Vleeschouwer, J. Meessen, C. Parisot, B. Macq, and J.-F. Delaigle, “Remote interactive browsing of videosurveillance content based on JPEG 2000,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 8, pp. 1143–1157, 2009.

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

[3] A. Mavlankar, D. Varodayan, and B. Girod, “Region-of-interest prediction for interactively streaming regions of high resolution video,” in *Proceedings of the 16th IEEE International Packet Video Workshop (PV ’07)*, Lausanne, Switzerland, November 2007.

[4] F. W. Mounts, “A video encoding system with conditional picture-element replenishment,” *Bell Systems Technical Journal*, vol. 48, no. 7, pp. 2545–2554, 1969.

[5] S. McCanne, M. Vetterli, and V. Jacobson, “Low-complexity video coding for receiver-driven layered multicast,” *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 6, pp. 983–1001, 1997.

[6] ISO/IEC 15444-1, JPEG 2000 image coding system, 2000.

[7] J. M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, 1993.

[8] D. Slepian and J. K. Wolf, “Noiseless coding of correlated information sources,” *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[9] R. Puri and K. Ramchandran, “PRISM: a new robust video coding architecture based on distributed compression principles,” in *Proceedings of the Allerton Conference on Communication, Control and Computing*, Allerton, Ill, USA, October 2002.

[10] A. Aaron and B. Girod, “Compression with side information using turbo codes,” in *Proceedings of the Data Compression Conference*, pp. 252–261, 2002.

[11] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero, “Distributed video coding,” *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71–83, 2005.

[12] X. Guo, Y. Lu, F. Wu, and W. Gao, “Distributed video coding using wavelet,” in *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 5427–5430, 2006.

[13] R. Bernardini, R. Rinaldo, P. Zontone, D. Alfonso, and A. Vitali, “Wavelet domain distributed coding for video,” in *Proceedings of IEEE International Conference on Image Processing*, pp. 245–248, 2006.

[14] A. Wang, Y. Zhao, and L. Wei, "Wavelet-domain distributed video coding with motion-compensated refinement," in *Proceedings of IEEE International Conference on Image Processing*, pp. 241–244, 2006.

[15] Y. Tonomura, T. Nakachi, and T. Fujii, "Distributed video coding using JPEG 2000 coding scheme," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 90, no. 3, pp. 581–589, 2007.

[16] N.-M. Cheung and A. Ortega, "Compression algorithms for flexible video decoding," in *Visual Communications and Image Processing*, vol. 6822 of *Proceedings of SPIE*, San Jose, Calif, USA, January 2008.

[17] R. G. Gallager, *Low Density Parity Check Codes*, Research Monograph Series, no. 21, MIT Press, Cambridge, Mass, USA, 1963.

[18] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Transactions on Communications*, vol. 44, no. 9, pp. 1261–1271, 1996.

[19] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

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

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

[22] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158–1170, 2000.

[23] D. J. C. MacKay, "Fountain codes," *IEE Communications*, vol. 152, no. 6, pp. 1062–1068, 2005.

[24] M. Luby, "LT codes," in *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pp. 271–280, 2002.

[25] A. Shokrollahi, "Raptor codes," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, pp. 2551–2567, 2006.

[26] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, 1999.

[27] J.-L. Kim, U. N. Peled, I. Perepelitsa, V. Pless, and S. Friedland, "Explicit construction of families of LDPC codes with no 4-cycles," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2378–2388, 2004.

[28] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, 2001.

[29] C. A. Cole, S. G. Wilson, E. K. Hall, and T. R. Giallorenzi, "A general method for finding low error rates of LDPC codes," *Computing Research Repository*, vol. 2006, Article ID 0605051, 2006.

[30] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 9, pp. 1445–1453, 1988.

[31] A. Ortega, "Optimal bit allocation under multiple rate constraints," *Proceedings of the Data Compression Conference*, pp. 349–358, April 1996.

[32] A. Ortega, K. Ramchandran, and M. Vetterli, "Optimal trellis-based buffered compression and fast approximations," *IEEE Transactions on Image Processing*, vol. 3, no. 1, pp. 26–40, 1994.

[33] F.-O. Devaux, C. De Vleeschouwer, and L. Schumacher, "A flexible video server based on a low complex post-compression rate allocation," in *Proceedings of the 16th International Packet Video Workshop (PV '07)*, Lausanne, Switzerland, November 2007.

[34] F.-O. Devaux, *JPEG 2000 and parity bit replenishment for remote video browsing*, Ph.D. thesis, September 2008.

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

[36] D. Schonberg, C. Yeo, S. C. Draper, and K. Ramchandran, "On compression of encrypted video," *Proceedings of the Data Compression Conference*, pp. 173–182, 2007.

[37] M. Fresia and G. Caire, "A practical approach to lossy joint source-channel coding," submitted to *IEEE Transactions on Information Theory*.