

Research Article

A Low-Complexity UEP Methodology Demonstrated on a Turbo-Encoded Wavelet Image Satellite Downlink

Eric Salemi,^{1,2,3} Claude Desset,^{1,3} Antoine Dejonghe,^{1,3} Jan Cornelis,^{1,2,3} and Peter Schelkens^{1,2,3}

¹ Interuniversity Microelectronics Center (IMEC), Kapeldreef 75, B-3001 Leuven, Belgium

² Vrije Universiteit Brussel (VUB), Faculty of Applied Science, Department ETRO, Pleinlaan 2, B-1050 Brussel, Belgium

³ Interdisciplinary Institute for BroadBand Technology (IBBT), B-9050 Gent, Belgium

Correspondence should be addressed to Claude Desset, desset@imec.be

Received 1 March 2007; Revised 14 August 2007; Accepted 21 November 2007

Recommended by Dan Lelescu

Realizing high-quality digital image transmission via a satellite link, while optimizing resource distribution and minimizing battery consumption, is a challenging task. This paper describes a methodology to optimize a turbo-encoded wavelet-based satellite downlink progressive image transmission system with unequal error protection (UEP) techniques. To achieve that goal, we instantiate a generic UEP methodology onto the system, and demonstrate that the proposed solution has little impact on the average performance, while greatly reducing the run-time complexity. Based on a simple design-time distortion model and a low-complexity run-time algorithm, the provided solution can dynamically tune the system's configuration to any bitrate constraint or channel condition. The resulting system outperforms in terms of peak signal-to-noise ratio (PSNR), a state-of-the-art, fine-tuned equal error protection (EEP) solution by as much as 2 dB.

Copyright © 2008 Eric Salemi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

This paper focuses on an existing satellite transmission system based on a state-of-the-art joint source-channel coding solution, transmitting images from an orbital space module to an earth ground station through a classical DVB-S2 (digital video broadcast for satellite) channel. In this system, the FlexWave-II core [1–4] is the wavelet-based image coder providing embedded scalability and low computational complexity. In addition, the T@mpo [5, 6] provides an efficient low-latency low-power turbo coder enabling close-to-capacity performance.

Our purpose is to jointly optimize the source and channel cores to offer a reliable delivery of high-quality digital images. In order to maximize the end-user quality, the system should be flexible and able to dynamically select an optimal protection scheme, while meeting the bandwidth constraint and adapting to the varying channel conditions. Source scalability induces a sequential dependency and a natural unequal error sensitivity among the compressed source symbols. This phenomenon naturally calls for an unequal error protection (UEP) scheme allowing a gradual protection leveling as we move from important to unimportant symbols. UEP [7–12]

improves the system by protecting more the more important bits, and protecting less the less important bits, thus improving the average performance of the system with the same amount of resources.

Impairments occurring on transmission channels usually results in data erasure or data corruption. Corruption means that data may be received with errors, while erasure means that data is not received at all. A system transmitting data directly on the channel would likely undergo corruption. More complex system including an IP stack would internally handle the detection of errors, resulting in data erasure.

For erasure channels, techniques like priority encoding transmission (PET) [13] are generally used. The PET framework allows for an optimal distribution of the transmission bit budget R . Initially, many solutions were initially developed, based on dynamic programming (DP) algorithms [14–17]. Recent solutions using an initial rate-optimal optimization followed by a fast local search distortion-optimal or Lagrangian techniques [18–21] were developed to bring down the complexity to a linear $\mathcal{O}(R)$ order.

However, corruption channels require an error-detection step before the source decoder, in order to prevent error propagation. Classically, the source decoding is stopped after

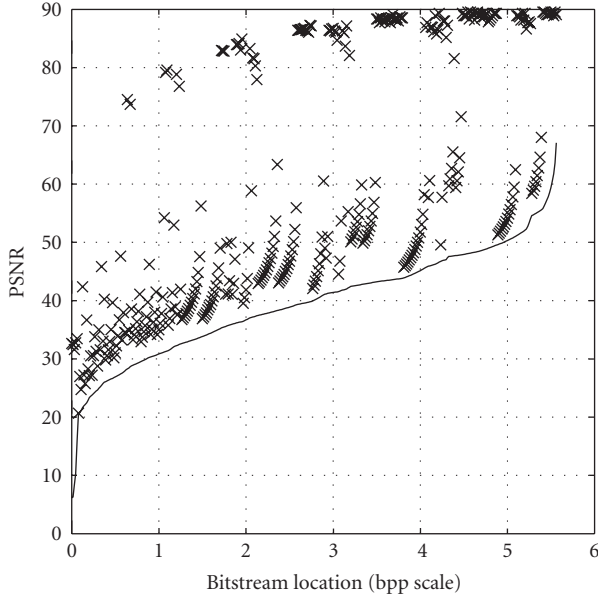


FIGURE 1: Reconstructed PSNR quality of a FlexWave-II bitstream when corrupted or truncated at specific locations.

the first detected errors, resulting in some parts of the transmitted content to be considered undecodable. Applied to the problem of joint source-channel optimization, various techniques like concatenated coding [22, 23], dynamic programming [23–25], exhaustive search [22], and gradient-based optimization [26, 27] are employed to solve different variants of the problem.

We note that all aforementioned techniques suppose that the source coder is not able to handle bitstream corruption, and somehow eliminate residual errors in order to feed the source decoding stage with uncorrupted data by either inserting an error detection stage, or by using packet-based transmission where the network itself suppresses residual errors by discarding data packets. This is suboptimal as recent coders have the possibility to efficiently use part of the data that was discarded. More specifically, by letting corrupted data enter the decoding stage, and building specific distortion models that evaluate the impact of corruption, we can optimize the system and exploit previously unused data.

As an example, we can see in Figure 1 the performance of FlexWave-II when the data is either truncated or corrupted at different locations in the bitstream. The x -axis represents the bitstream location on a bit per pixel scale, while the y -axis represents the PSNR quality obtained after decoding. The plain curve shows the PSNR quality when the bitstream is truncated. Each cross shows the PSNR quality when a single bit error is inserted, leaving the rest of the bitstream untouched. We can see that the distortion resulting from a bit error at any location in the bitstream is always smaller than the distortion resulting from a truncation at the same location. This means that the source decoder can efficiently use the data beyond the corruption point to reduce the distortion.

Our UEP methodology [28] proposes a novel, generic, and pragmatic approach to solve the source-channel allocation problem. It is based on a joint source-channel model that is steered at runtime by a low-complexity algorithm. This joint model is merging different models, respectively, characterizing the different components of the system (source, source coder, channel coder, and channel), and is enabled by a set of well-defined simplifying assumptions. These assumptions greatly reduce the complexity of the model. This joint source-channel model is actually very flexible, and is able to dynamically provide the rate-distortion characteristics of the system depending on parameters such as the global bit budget or the channel conditions. At runtime, these rate-distortion characteristics are exploited by a low-complexity algorithm that optimizes the code rate allocation. This paper focuses on the instantiation of our solution for the satellite communication system described before.

Because of complexity constraints, the source model is source-independent and only represents a statistical expectation of the rate-distortion behavior over a training set of satellite images. Hence, it is a priori suboptimal. Previous work [29] has demonstrated that the source-independent model had no significant impact on the end-to-end rate-distortion performance of our methodology.

In this paper, the UEP controller performance will be compared with a classical equal error protection (EEP) solution that simply utilizes the incoming order of the FlexWave-II bitstream as prioritization information. We will prove that the proposed UEP solution can dynamically adapt to varying transmission conditions, and outperforms the EEP scheme in the working range of channel conditions.

Section 2 gives an overview of our UEP methodology. Section 3 describes the general setup of the satellite communication system and derives the characteristics of the rate-distortion model. Section 4 shows the simulation results. Section 5 compares the simulated results to the performance of the hardware implementation. Section 6 concludes the paper.

2. UEP METHODOLOGY

The proposed generic UEP methodology can be incorporated in any system offering UEP capabilities. In previous work [28], this methodology has been successfully applied to a JPEG2000-based system. The goal of this paper is to apply the same methodology to a satellite compression system, and to demonstrate its performance.

Section 2.1 recalls the general problem statement. Section 2.2 deals with the joint modeling of the channel and source components. Sections 2.3 and 2.4, respectively, explain how the separate models are combined at run-time and how the resulting rate-distortion characteristics are exploited to derive the final protection allocation.

2.1. Problem statement

We consider the transmission of a scalable bitstream embedding S substreams. We have $P + 1$ discrete protection levels, including the possibility of transmitting a substream without

protection or not transmitting it at all. Protection levels are indexed from 0 to P , where 0 corresponds to the untransmitted case (cut substream), and 1 corresponds to the unprotected case (uncoded substream). A global bit budget R is available to transmit the data and is shared among these substreams. Our objective is to maximize the expected quality of the received data, or to minimize the expected distortion δ . Concerning the protection allocation, three important remarks have to be made.

The first remark is that the system allows residual bit errors in the transmitted substreams. This means that all substreams are effectively used by the source decoder, with a possibility to quality degradation when the source is reconstructed. The second remark is that each substream is considered as an independently decodable unit. This means that the amount of protection allocated to each substream (related to the amount of residual errors) can be independently and arbitrarily chosen. In other words, we are not constraining the resource distribution to be monotonically decreasing, as would be done in the case of a progressive bitstream [22, 30].

It could be argued that even though a scalable bitstream is not necessarily progressive, decoding dependencies may subsist in the bitstream. Actually, this decoding dependency is the cause of the unequal error sensitivity observed in a scalable bitstream. Additionally, the proposed solution measures this error sensitivity through a model and unequally distributes the protection accordingly. Therefore, the joint source-channel model is a central tool that allows the algorithm to gradually match the protection level to the error sensitivity and thus taking into account the possible decoding dependencies.

We assume the total expected image distortion δ to be the sum of the expected distortion for each image substream [31]. This is expressed by the following equation:

$$\delta(\psi) = \sum_{1 \leq s \leq S} \delta_s(P_s), \quad (1)$$

where ψ represents the S -tuple (P_1, \dots, P_S) of protection levels applied, respectively, to the S substreams; and $\delta_s(P_s)$ is the distortion contribution of substream s associated with protection level P_s . Given a protection set ψ we compute the global rate required:

$$\rho(\psi) = \sum_s \rho_s(P_s) = \sum_{P_s \neq 0} \frac{L_s}{R(P_s)}, \quad (2)$$

where L_s is the length of substream s and $R(P_s)$ is the channel coding rate for the protection P_s . Smaller coding rates give better protection levels and increase the corresponding rate expense. Protection $P_s = 0$ incurs no rate expense since the corresponding substream data will not be transmitted. The problem is solved by finding the optimal protection set $\hat{\psi}$ that minimizes the global distortion $\delta(\psi)$, while meeting the global rate constraint $\rho(\psi) \leq R$:

$$\hat{\psi} = \arg \min_{\psi} \delta(\psi) \quad \text{s.t.} \quad \rho(\psi) \leq R. \quad (3)$$

This additive distortion model allows for an independent optimization of the protection levels for each substream, and thus greatly simplifies the task of the runtime optimization. In the following, we give more details about the distortion model.

2.2. Joint source-channel distortion model

The joint source-channel distortion model is actually a combination of two simpler models which individually estimate the characteristics of the source coder and the different protection modes of the channel coder. This section describes the computation of the individual source and channel models, and explains how they are combined into the joint source-channel model.

2.2.1. Source model

The source model evaluates the distortion induced by cutting or corrupting individual substreams. This is done in two steps.

- (i) First, we compute the S values D_s^{cut} , which represent the MSE distortion resulting after cutting the substream s out of the bitstream while leaving other substreams untouched. It should be noted that cutting substream s means that protection level $P_s = 0$ has been assigned to substream s .
- (ii) Secondly, we compute the S values D_s^{bit} which estimate the average MSE distortion per erroneous bit in the substream s . This is obtained by inserting individual bit errors in the substream, while leaving remaining bits uncorrupted.

2.2.2. Channel model

The channel coder offers P distinct protection levels. Depending on the channel quality q and the protection level p , the channel model provides an average estimation of the bit error rate (BER), which we will denote $b(q, p)$.

2.2.3. Joint source-channel model

Considering a fixed channel quality q , the joint source-channel model estimates the expected MSE distortion $\delta_s(p)$ inside substream, depending on the protection level p . Since residual errors are considered independent, we can simply estimate the distortion $\delta_s(p)$ in function of the estimated residual BER $b(q, p)$. To obtain a usable model, we estimate the expected MSE distortion $D_s^{\text{ber}}(b)$ on the range of possible BER values b between 0 and 0.5. To achieve that, we simply measure D_s^{ber} on a discrete set of BER values, relying on a linear extrapolation for intermediate values.

It should be noted that when the residual BER within substream s is equal to $1/L_s$, the average number of errors is equal to 1, and the expected MSE distortion $D_s^{\text{ber}}(1/L_s)$ is matching the average bit distortion D_s^{bit} . Eventually, we are able to estimate the expected distortion within substream s ,

undergoing loss or corruption according to the following equations:

$$\begin{aligned}\delta_s(0) &= D_s^{\text{cut}}, \\ \delta_s(P_s) &= D_s^{\text{ber}}[b(q, P_s)].\end{aligned}\quad (4)$$

2.3. Rate-distortion curves

Consider the transmission of a bitstream with protection $p - 1$. Assuming the substream, s has its protection level upgraded from $p - 1$ to p , we express the distortion reduction $\Delta_{s,p}$ as

$$\Delta_{s,p} = \delta_s(p - 1) - \delta_s(p). \quad (5)$$

Hence, the distortion reduction has been evaluated as if the substream with protection $p - 1$ was cut from the bitstream and added again with protection p . Rewriting (5) for the case when the substream s is simply added to the bitstream delivers

$$\Delta_{s,1} = \delta_s(0) - \delta_s(1) = D_s^{\text{cut}} - \delta_s(1). \quad (6)$$

Furthermore, we define the importance value $I_{s,p}$ as the ratio between the distortion decrease and the bitrate increase induced by upgrading the protection level of the substream s from $p - 1$ to p :

$$I_{s,p} = \frac{\Delta_{s,p}}{(1/R(p) - 1/R(p - 1))L_s}. \quad (7)$$

Actually, the set of importance values $I_{s,p}$ matches exactly the slope values of the rate-distortion curve for substream s . We assume here that the obtained rate-distortion curve is convex. However, if this is not the case, we can prune out protection levels for a specific substream so that the $I_{s,p}$ slope series is monotonically decreasing. At most, $I_{s,p}$ values must be computed for all possible protection levels p from 1 to P and for all substreams s from 1 to S . It yields a maximum number of PS importance values.

2.4. Proposed runtime algorithm

According to (7), we have at most $K = PS$ importance values $I_{s,p}$, with $1 \leq s \leq S$ and $1 \leq p \leq P$. $I_{s,p}$ represents the relative importance or quality improvement that would be observed if the protection level of substream s would be upgraded to p . This actually means that these importance values represent the slopes of the rate-distortion curves associated to each associated to the S substreams.

These K values are now sorted in decreasing order and the corresponding indices are arranged in two series (s_k) and (p_k). The allocation is done with an iterative process over the K stages. At stage $k = 0$, all substreams are initialized to $p = 0$. At each stage k , the substream s_k is upgraded to protection level p_k until we reach stage $k = PS$, where all substreams are maximally protected with protection level P .

As an example, in Figure 2 we have $S = 2$ substreams, $P = 3$ protection levels, and $K = 6$ importance values. We see that

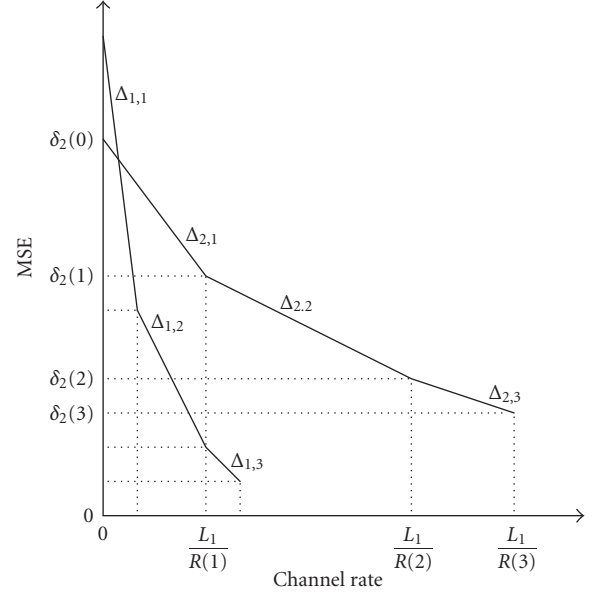


FIGURE 2: An example of rate-distortion characteristics obtained with 2 substreams and 3 protection levels.

TABLE 1: Protection levels allocation of the proposed algorithm, corresponding to the rate-distortion characteristics of Figure 2.

Stage	1	2	3	4	5	6
Substream 1	0	1	2	2	3	3
Substream 2	0	0	0	1	2	3

the importance values are sorted in the following decreasing order: $\Delta_{1,1}$, $\Delta_{1,2}$, $\Delta_{2,1}$, $\Delta_{1,3}$, $\Delta_{2,2}$, and $\Delta_{2,3}$. Table 1 shows how the proposed UEP algorithm attributes the protection levels to the 2 substreams in a 6-stage allocation.

During the algorithm, we also form the series of protection set (ψ_k) and rate expense (ρ_k). ψ_0 is the protection set where all substreams are cut. ρ_0 is therefore equal to 0 since no substream is transmitted. ψ_k is defined follows:

$$\psi_k = (P_1^k, \dots, P_{s_k}^k, \dots, P_S^k), \quad (8)$$

where P_s^k is the protection level associated with substream s at stage k . We derive ψ_k from ψ_{k-1} by upgrading the protection level of substream s_k to p_k . Therefore, ψ_k is identical to ψ_{k-1} except for its s_k th element, which is equal to p_k . Accordingly, we derive ρ_k from ρ_{k-1} by adding the extra rate incurred by protection p_k on substream s_k . Using (2), we define the global rate ρ_k :

$$\begin{aligned}\rho_k &= \rho(\psi_k) = \sum_{s \neq s_k} \frac{L_s}{R(P_s^k)} + \frac{L_{s_k}}{R(P_{s_k}^k)} \\ &= \rho_{k-1} - \frac{L_{s_k}}{R(P_{s_k}^{k-1})} + \frac{L_{s_k}}{R(P_{s_k}^k)}.\end{aligned}\quad (9)$$

We eventually obtain the rate sets (ρ_k) and the corresponding optimal protection sets (ψ_k). Thanks to the reordering operation, the global optimization is achieved by selecting

the highest ρ_k being smaller than the target rate R . After the global optimization step, the two series (ρ_k) and (ψ_k) enable the system to reach an optimal protection set for any rate constraint. This means that our low-complexity algorithm is very dynamic and can adapt to any rate condition with a simple search, without loss of optimality in the specific case of convex rate-distortion characteristics.

2.5. Complexity evaluation

The computation of the importance values $\Delta_{s,p}$ in (7) requires $K = 3PS$ multiplications and $2K$ additions, according to (7), and (4). The sorting costs an expected $K \log_2(K)$ comparisons. The (ψ_k) series computation do not require any computation. According to (9), each ρ_k computation needs 1 multiplication and 2 additions for a total of K multiplications and $2K$ additions. The selection of the optimal \hat{k} is performed by a bisection search and requires an expected $\log_2(K)$ comparisons in order to find the optimal \hat{k} . If we consider that the multiplication is the dominant term, the proposed algorithm has a complexity of order $\mathcal{O}(PS)$, which is linear with respect to the number of substreams S and the number of protection levels P . Given that the number of protection levels can be limited to 3, the proposed runtime algorithm has a very low complexity.

3. SYSTEM SETUP

The transmission of the data from the satellite to the ground station is performed over a DVB-S2 channel. Basically, the FlexWave-II still image encoder produces a progressive bitstream by outputting a series of data substreams that holds a varying number of bytes. These substreams are forwarded to the T@mpo encoder that adds a certain number of parity symbols depending on the selected protection mode. The protected substreams are then sent directly on the transmission channel and received by the T@mpo decoder. The decoded substreams are then fed to the FlexWave-II decoder, which subsequently decodes the image.

3.1. Source

Since satellite imaging is targeted, it is therefore necessary to optimize the source model for this application. To this end, we chose the black and white version of the Toulouse image represented in Figure 3.

The main advantage of the methodology [28] is the separation of the design-time modeling phase and the runtime optimization phase. In the ideal case, the source model is perfectly matching the distortion characteristic of the transmitted image. However, this can only be obtained by computing the model at runtime, which is unpractical given the high complexity of the modeling process. A real-life transmission system will therefore utilize a model calculated offline based on a training set of images, which we address as the source-independent model. When a communication system is transmitting a specific class of images like space imagery as our satellite data, the source-independent model will be statisti-



FIGURE 3: The Toulouse image (512 × 512 pixels, 8 bits per pixel).

cally close to the type of images that are being transmitted, as proved in the next paragraph.

The distortion characteristics of the source-independent model are based on a training set of I images: we first compute the IS components $D_{i,s}^{\text{cut}}$, $D_{i,s}^{\text{bit}}$, and $D_{i,s}^{\text{ber}}$ as described in Section 2, which correspond to the I individual source models for each training image. We obtain the S source-independent model components $\widehat{D}_s^{\text{cut}}$, $\widehat{D}_s^{\text{bit}}$, and $\widehat{D}_s^{\text{ber}}$ by averaging the individual models over the training set.

Two source models are computed. The reference source model is directly computed from the Toulouse image itself. The source-independent model training set contains 12 images that were taken from the USC-SIPI free image database [32]. It represents an average source model for satellite image class. Further on in this document, we refer, respectively, to these models as *Toulouse* and *Sipi* models.

From the series of $\widehat{D}_s^{\text{cut}}$ values, it is natural to sort the substreams by decreasing distortion values. Conceptually, the bitstream order is a property, which is only dependent on the characteristics of the source coder and, therefore, we only use the cut distortion values $\widehat{D}_s^{\text{cut}}$. As we averaged the distortions characteristics of each substream over a set of training images, we obtain a probabilistic importance order of the substreams, which we call the source-independent bitstream order.

Figures 4 and 5 show a comparison of the distortion characteristics between the Toulouse model and the Sipi model. On both curves, the x -axis is the substream index, following the source-independent bitstream order, and the y -axis represents the MSE distortion. The plain curve represents the Sipi model. The dashed curve follows the Toulouse model profile. The Sipi model matches well the Toulouse model, apart from some local deviations. This is a logical conclusion since the Sipi model is based on a training set of images that represent specifically the class of images to which Toulouse belongs.

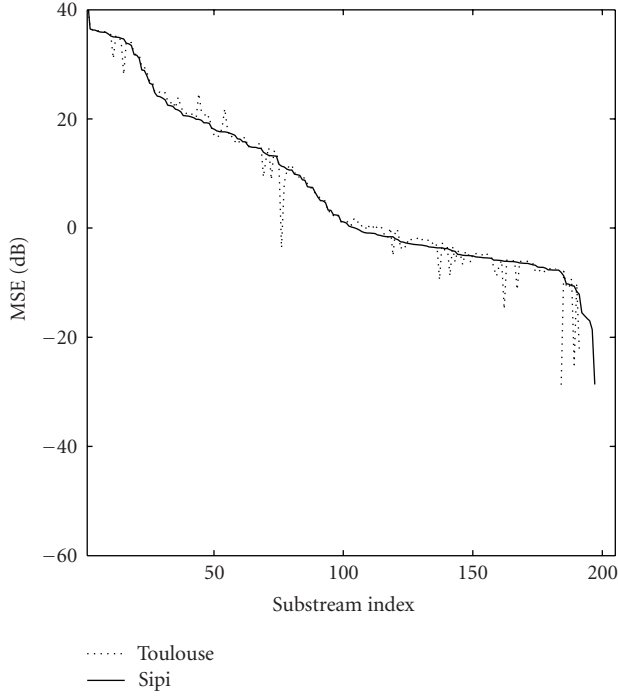


FIGURE 4: Toulouse D^{cut} and Sipi \widehat{D}^{cut} source model distortion profile.

3.2. Source coder

The source coder used in this satellite system is based on the FlexWave-II architecture. This architecture has been specifically designed as a dedicated compression component for space-born applications. It is based on a 9/7 wavelet decomposition, which is also used by similar state-of-art source coders like SPIHT [33] and JPEG2000 [34]. However, the SPIHT and JPEG2000 are fully featured source coders that are too complex to implement in a low-power cost-efficient application specific integrated circuit (ASIC) realization for space applications. Therefore, specific algorithmic simplifications have been brought to the FlexWave-II core in order to reduce the complexity of the solution at the cost of a slight compression performance decrease. On a field-programmable gate array (FPGA) implementation of the FlexWave-II, clocked at 41 MHz, a processing performance of up to 10 Mpixels/s was measured. For this paper, we configured the FlexWave-II core for a 4-level wavelet decomposition depth, which outputs a total of $S = 349$ substreams.

3.3. Channel

Typically, the quality of service offered over a DVB-S2 channel is subject to tropospheric phenomena, such as rain and clouds, as well as the influence of atmospheric gas. Both can severely degrade the quality of the transmission channel. These effects can have an influence on the long-term distribution of the channel attenuation statistics.

Figure 6 represents a simulated time series of $N = 7200$ samples for a typical DVB-S2 channel. The channel simulator is outputting correlated channel coefficients at a basic

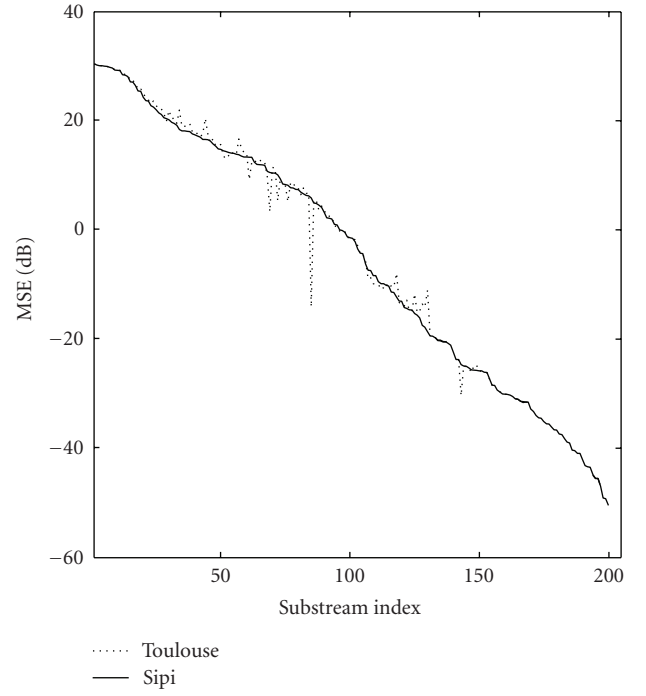


FIGURE 5: Toulouse D^{bit} and Sipi \widehat{D}^{bit} source model distortion profile.

frequency $F_c = 2$ Hz, so that the channel series spans over 1 hour. The actual datarate of the system is $R_s = 45$ Mbit/s. Therefore, we can insert approximately 2.8 Mbytes of data between 2 consecutive samples. Considering a standard size compressed picture to be sent on this channel, we see that it will be entirely contained between two consecutive coefficients. Moreover, due to the time-domain correlation, two consecutive samples will have similar amplitudes (see Figure 6). As a consequence, we can already anticipate that the system will exclusively work in slow fading mode. This means that the protection allocation optimizer can safely consider the channel as a constant additive white Gaussian noise (AWGN) channel with a specific signal-to-noise ratio for the complete transmission of an image corresponding to the current attenuation of the DVB-S2 channel.

In the remainder of the document, we will therefore focus on the end-to-end performance of the system over an AWGN channel. The derivation of the performance over the DVB-S2 channel is simply performed by a convolution between an AWGN performance curve and the modeled DVB-S2 channel statistic profile.

3.4. Channel coder

The channel coder used in the T@mpo system is an efficient implementation of a low-latency low-power turbo coder/decoder based on parallel concatenated convolutional turbo codes (PCCC). The T@mpo coder has 4 protection modes allowing the system to adapt the degree of protection against errors. The protection levels are described by their respective coderates in Table 2.

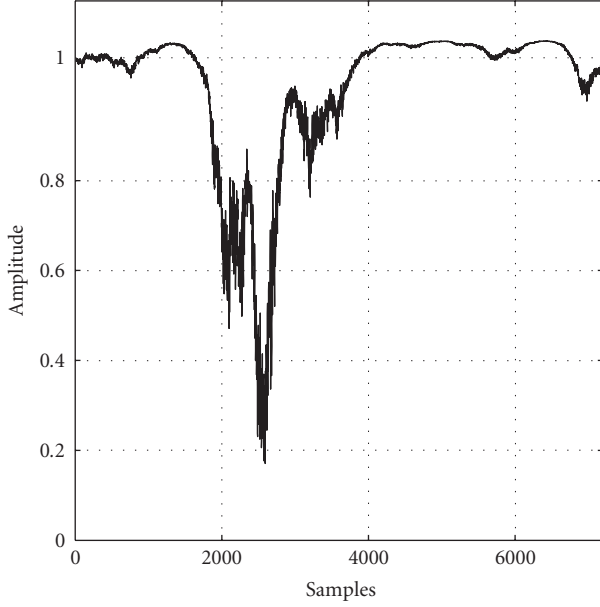


FIGURE 6: DVB-S2 channel time series.

TABLE 2: Available protection levels for the T@mpo channel coder.

Protection level	Coderate
1 (strongest)	1/3
2	1/2
3	2/3
4 (weakest)	3/4

Under independent channel errors assumption [28], the BER after decoding is taken as the only parameter to characterize the occurrence of errors in the system. In Section 3.3 we considered that computing the performance of the system transmitting over AWGN channels was sufficient to accurately derive the performance of the system over the considered DVB-S2 satellite channel. Figure 7 gives an overview of the performance of the T@mpo channel coder over an AWGN channel. The x -axis represents the signal-to-noise ratio E_s/N_0 , while the y -axis represents the BER at the output of the channel decoder. Plain curves represent the performance of the 4 modes of the T@mpo coder as presented in Section 3.4. The dashed curve represents the classical non-coded performance on an AWGN channel.

4. SIMULATIONS

In this section, we compare the performance of the full UEP controller with an EEP controller that would equally protect the bitstream with a single average protection level. As introduced in Section 2, a predictive model of the end-to-end distortion propagation is required by the full UEP controller in order to optimize the protection allocation. This predictive model is based on the assumption that the distortion caused by transmission errors is additive at the substream level. This approximation is required to enable the low-complexity optimization described in Section 2.4, but may introduce a mis-

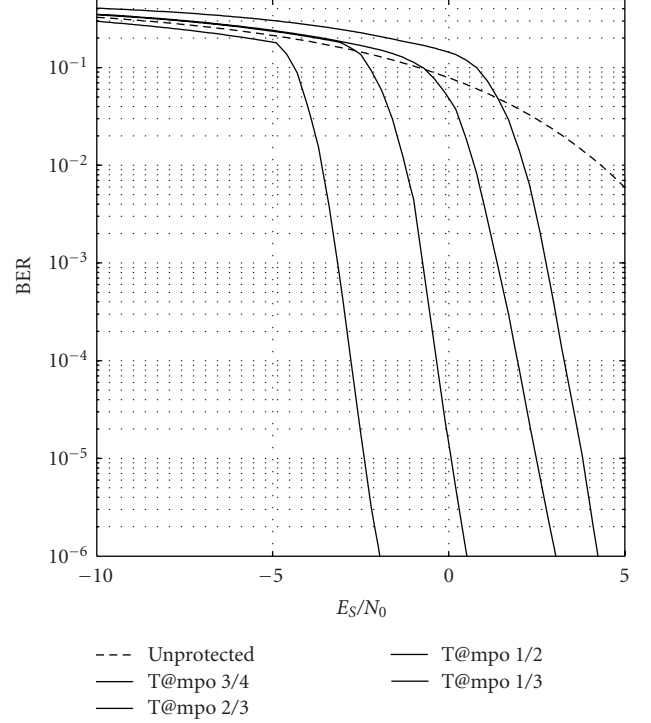


FIGURE 7: BER performance of the T@mpo channel coder on an AWGN channel.

match between the estimated distortion during the optimization of the protection allocation and the actual distortion observed at the receiver. Depending on the amount of mismatch, the performance of the UEP allocation may be deteriorated.

Though, the parameters of the simulations have been previously introduced in Section 3, they are briefly recalled hereafter. The number of encoded substreams is $S = 349$ and corresponds to a 4-level wavelet decomposition. The number of protection levels is equal to $P + 1 = 6$, and accounts for the 4 T@mpo protection modes (see Table 2) plus the additional unprotected and nontransmitted modes. It was shown in the literature [35] that three protection levels are usually sufficient to obtain most UEP gains for binary symmetric channels with error probabilities inferior to 10^{-1} . Therefore, our system used a sufficiently high number of protection levels.

In what follows, we compare the simulated end-to-end performance of our solution with a state-of-the-art EEP solution and assess the impact of the additivity assumption on the end-to-end performance.

4.1. End-to-end performance

In this section, we compare the end-to-end-performance of the proposed UEP controller with that of an advanced EEP system. A general EEP algorithm simply utilizes the order of the embedded substreams as prioritization information. The image is encoded by the source coder, which subsequently outputs an ordered sequence of substreams. The substreams are further protected by the channel coder with a

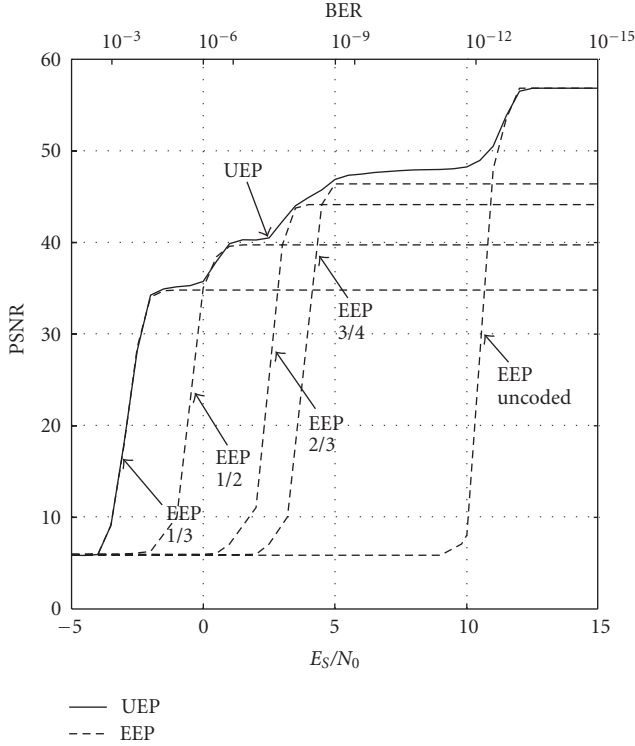


FIGURE 8: Performance comparison between UEP and optimized EEP for the transmission of Toulouse.

single error correcting code until the bit budget is exhausted. The remaining part of the bitstream is discarded and therefore not transmitted. Note that such an EEP solution relies already on a progressive bitstream, which can be cut at any place and is provided in a rate-distortion optimized order.

Figure 8 compares the performance of the EEP and UEP controllers for a global budget corresponding to the size of the Toulouse source bitstream. The plain curve shows the performance of the UEP controller, while the dashed curve shows the PSNR performance of the EEP controller. In a classical EEP system, the protection level is fixed for the whole range of channel conditions. In this simulation, the EEP performance is actually derived from the hull of all possible EEP optimization, given the number of protection levels available in the system. Therefore, the EEP performance of Figure 8 corresponds to an EEP controller that would choose the optimal protection mode according to the channel condition. It should be noted that a classical EEP system cannot achieve such an optimization since the protection level is fixed. However, for the UEP controller, the allocation is based on a predictive model, which is directly dependent on the channel condition. Therefore, the protection levels are automatically adapted prior to transmission.

The bottom x -axis represents the signal-to-noise ratio E_s/N_0 , while the top x -axis represent the equivalent uncoded BER on an AWGN with binary phase-shift keying (BPSK) modulation. For low and high E_s/N_0 , the performance of both the EEP and the UEP controller are closely matched. This is explained by the fact that for E_s/N_0 below -3 dB and

above 12 dB, single protection modes are selected by both algorithms. Looking at Figure 7, we see that for bad channel conditions ($E_s/N_0 = -3$ dB), the best T@mpo mode (1/3 rate) gives a BER of 10^{-3} while the next best mode (1/2 rate) gives a BER above 0.1. Both algorithms decide to transmit 1/3 of the bitstream with the best T@mpo mode. Similarly, for very good channel conditions ($E_s/N_0 > 12$ dB), the unprotected mode is subject to a sufficiently low BER to deliver the whole bitstream without any protection. For intermediate channel conditions (E_s/N_0 between -2 dB and 12 dB), the image reconstruction quality is acceptable, with a PSNR above 30 dB and the UEP controller outperforms the EEP controller by as much as 2 dB.

It should be noted that for both controllers, the reconstructed quality has a staircase effect. This effect is clearly visible on the EEP performance curve. The different switching points actually correspond to the channel conditions where the EEP controller decides to switch to the next protection mode. This effect is mainly due to the fact that the number of protection levels is limited. Indeed, for each protection level, only one bitstream truncation point is possible in order to fit the available budget. Between consecutive switching points, the amount of source data will therefore be constant and correspond to a quality plateau. At the next protection mode switch, the truncation point jumps further along the bitstream. Looking at the UEP controller performance, we remark that the staircase effect is less visible, giving a smoother transition between the switching points. This is explained by the fact that the UEP controller can allocate multiple protection rates across the substreams and trade more precisely source and channel resources for a given channel condition. It should be stressed that the UEP controller automatically adapts the number of protection levels used and their distribution across the substreams according to the algorithm described in Section 2.4.

4.2. Impact of additivity mismatch

The additivity assumption is central to the optimization algorithms proposed in [28] and in Section 2. It allows the use of a low-complexity algorithm for the UEP global optimization. First, we characterize the amplitude of the mismatch with large parameters $P + 1 = 6$ and $S = 349$ in order to characterize the deviation for the system setup described in Section 3. In a second step, we evaluate the end-to-end performance and the mismatch for small parameters $P = 2$ and $S = 2$. The impact of the deviation on the end-to-end is actually checked against a reference full-search algorithm, which is only feasible when the parameters are small. Since the deviation has no impact when parameters are small, and that deviation characteristics are similar whether we use small or large parameters, we suppose that the system will keep good performance with large parameters. Details of the simulations are given hereafter.

Uniform BERs ranging from 10^{-6} to 10^{-1} are applied on the different substreams. For each BER, 100 simulations are run to obtain a reasonable averaging of the MSE and the peak signal-to-noise ratio (PSNR) measurements. First we jointly corrupt all substreams with a fixed BER and compute the

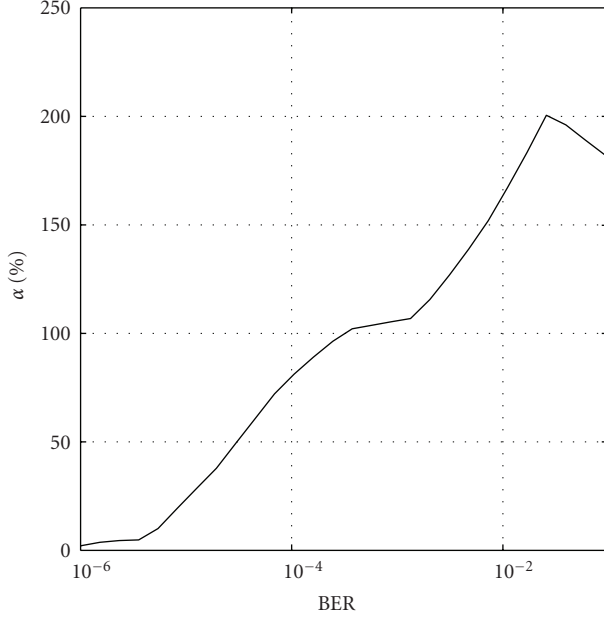


FIGURE 9: Additivity mismatch for Toulouse, defined as excess in total expected distortion (using the additive model) over the simulated overall distortion (true value), as a function of BER.

output distortion δ . Secondly, we corrupt each of the S substreams with a fixed BER b while leaving other substreams uncorrupted, and compute the S individual distortions D_s , where $1 \leq s \leq S$. Figure 9 shows the additivity mismatch defined as

$$\alpha = 1 - \frac{\delta}{\sum_{s=1}^S D_s}, \quad (10)$$

which happens to be strictly positive. This confirms that the additivity-based distortion estimation overestimates the real joint distortion. The mismatch starts off with less than 10% mismatch at a BER of 10^{-5} and reaches a plateau at 100% for a BER of 10^{-3} before reaching a peak at 200% for a BER of 3×10^{-2} . Clearly additivity is not respected within FlexWave-II and exhibits a large additivity deviation. However, it should be stressed that a model mismatch does not necessarily lead to a wrong decision during the optimization phase or a decrease in the end-to-end performance of the system.

To assess the impact of the additive model deviation on the end-to-end performance, we compared the output optimization decision with a full-search algorithm. A full-search algorithm basically computed the expected distortion of all possible protection allocations prior to the transmission, and picked the best allocation based on the lowest distortion value. The full-search algorithm is not realizable with the large parameters $P + 1 = 6$ and $S = 349$ used in Section 4.1. However, with $P + 1 = 3$ and $S = 2$, we found that the protection allocation performed by the system with the additive model was identical to that of the full-search algorithm, while having similar mismatch amplitudes. Therefore, we assume that the behavior of our low-complexity solution will remain optimal with increasing parameters.

As a final comment, we have to state that the UEP algorithms optimally match the protection levels to the importance of each substream. By increasing the protection of important substreams, we expect to reduce their large contribution to the distortion. Hence, we expect UEP to mitigate the masking effect [31] when the parameters S and P are increased, which is one of the main cause for the additivity mismatch, as dominant substreams will be heavily protected.

5. HARDWARE IMPLEMENTATION

During the development of the satellite communication system, a hardware implementation of the UEP-optimized system has been realized. This section briefly describes the hardware setup that was designed. The hardware platform has been realized on a PICARD system www.imec.be/wireless/picard. The PICARD system consists of a PC in an industrial 19-inch rack. The backplane of the rack exposes a compact PCI (C-PCI) backplane. On this backplane, boards containing IP cores can be plugged. The T@mpo, FlexWave-II and AWGN channel are all integrated on such a circuit board. The board is built around a central FPGA that interconnects all the IP cores.

Figure 10 shows the comparison between the software version of the system presented in Section 4.1, and the hardware platform that has been instantiated. The transmission scenario described in Section 4 is used. The plain curve of Figure 10 is therefore identical to the plain curve of Figure 8, showing the performance of the UEP controller. The starred curve shows the performance of the Hardware implementation. As we can see, there is an almost perfect match between the two curves. This validates the hardware implementation of the FlexWave-II and T@mpo cores compared to their software versions. A processing performance of up to 10 Mpixels/s was measured on the final platform.

6. CONCLUSIONS

We have shown that joint source-channel optimization is a promising technique for the future of satellite imaging. By combining the embedded scalability offered by state-of-the-art wavelet-based source coders and recent channel coding techniques that are providing a flexible range of protection levels, and applying a generic UEP methodology on the combined system, we have developed an efficient satellite image transmission system. The proposed UEP solution outperforms an optimized state-of-the-art EEP solution by as much as 2 dB in the working range of channel conditions, and is able to adapt to any bitrate and any channel condition. The inherent low complexity of the resulting solution, enabled by an efficient joint source-channel modeling of the system, allowed the practical implementation of the complete system on a hardware platform and proved to have a rate-distortion performance very close to the software platform.

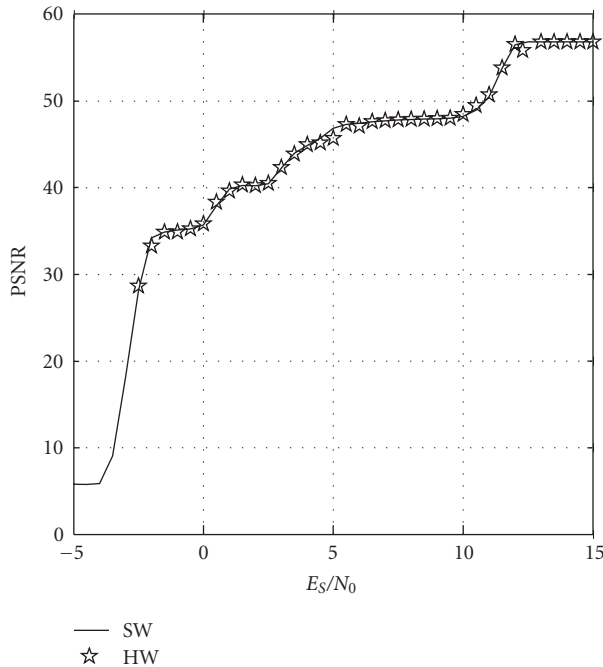


FIGURE 10: Performance comparison between software and hardware implementation for the transmission of Toulouse.

ACKNOWLEDGMENTS

The authors would like to thank the IMEC TOTEM team for the development of the software and hardware platform as well as for the majority of the results produced for this paper. Peter Schelkens was supported by a postdoctoral mandate of the Fund for Scientific Research—Flanders (FWO). This work has been funded and supported by the European Space Agency (ESA) through the Tandem Optimized Turbo Encoded Multimedia (TOTEM) project.

REFERENCES

- [1] L. Nachtergaele, J. Bormans, G. Lafruit, B. Vanhoof, and I. Bolsens, "Methodological reduction of memory requirements for a vlsi spaceborne wavelet compression engine," in *Proceedings of the 6th International Workshop on Digital Signal Processing Techniques for Space Applications (DSP '98)*, pp. 3–4, Noordwijk, The Netherlands, September 1998.
- [2] p. Schelkens, G. Lafruit, F. Decroos, J. Cornelis, and F. Catthoor, "Power exploration for embedded zero-tree wavelet encoding," in *Proceedings of International Symposium on Low Power Electronics and Design (ISLPED '99)*, San Diego, Calif, USA, August 1999.
- [3] B. Masschelein, J. G. Bormans, and G. Lafruit, "The local wavelet transform: a cost-efficient custom processor for space image compression," in *Applications of Digital Image Processing XXV*, vol. 4790 of *Proceedings of SPIE*, pp. 334–345, Seattle, Wash, USA, July 2002.
- [4] B. Vanhoof, B. Massachelein, A. Chirila-Rus, and R. Osorio, "The FlexWave-II: a wavelet-based compression engine," in *European Space Components Conference (ESCCON '02)*, pp. 301–308, Toulouse, France, December 2002.
- [5] A. Giulietti, B. Bougard, V. Derudder, S. Dupont, J.-W. Weijers, and L. Van der Perre, "A 80 Mb/s low-power scalable turbo codec core," in *Proceedings of the Custom Integrated Circuits Conference*, pp. 389–392, Orlando, Fla, USA, May 2002.
- [6] B. Bougard, A. Giulietti, V. Derudder, et al., "A scalable 8.7nJ/bit 75.6Mb/s parallel concatenated convolutional (turbo) codec," in *IEEE International Solid-State Circuits Conference, Digest of Technical Papers (ISSCC '03)*, vol. 1, pp. 152–484, San Francisco, Calif, USA, February 2003.
- [7] R. Hamzaoui, V. Stanković, and Z. Xiong, "Fast joint source-channel coding algorithms for internet/wireless multimedia," in *Proceedings of International Joint Conference on Neural Networks (IJCNN '02)*, vol. 3, pp. 2108–2113, Honolulu, Hawaii, USA, May 2002.
- [8] A. Natsu and D. Taubman, "Unequal protection of JPEG2000 code-streams in wireless channels," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '02)*, vol. 1, pp. 534–538, Taipei, Taiwan, November 2002.
- [9] K. Joohee, R. M. Mersereau, and Y. Altunbasak, "Error-resilient image and video transmission over the Internet using unequal error protection," *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 121–131, 2003.
- [10] H. Cai, B. Zeng, G. Shen, and S. Li, "Error-resilient unequal protection of fine granularity scalable video bitstreams," in *Proceedings of the IEEE International Conference on Communications (ICC '04)*, vol. 3, pp. 1303–1307, Paris, France, June 2004.
- [11] Z. Wu, A. Bilgin, and M. W. Marcellin, "Joint source/channel coding for image transmission with JPEG2000 over memoryless channels," *IEEE Transactions on Image Processing*, vol. 14, no. 8, pp. 1020–1032, 2005.
- [12] Z. Liu, M. Zhao, and Z. Xiong, "Efficient rate allocation for progressive image transmission via unequal error protection over finite-state Markov channels," *IEEE Transactions on Signal Processing*, vol. 53, no. 11, pp. 4330–4338, 2005.
- [13] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1737–1744, 1996.
- [14] R. Puri and K. Ramchandran, "Multiple description source coding using forward error correction codes," in *Proceedings of the 33rd Asilomar Conference on Signals, Systems, and Computers*, vol. 1, pp. 342–346, Pacific Grove, Calif, USA, October 1999.
- [15] A. E. Mohr, E. A. Riskin, and R. E. Ladner, "Approximately optimal assignment for unequal loss protection," in *Proceedings of International Conference on Image Processing (ICIP '00)*, vol. 1, pp. 367–370, Vancouver, BC, Canada, September 2000.
- [16] T. Stockhammer and C. Buchner, "Progressive texture video streaming for lossy packet networks," in *Proceeding of the 11th International Packet Video Workshop (PV '01)*, p. 57, Kyongju, Korea, May 2004.
- [17] S. Dumitrescu, X. Wu, and Z. Wang, "Globally optimal uneven error-protected packetization of scalable code streams," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 230–239, 2004.
- [18] R. Puri, K. -W. Lee, K. Ramchandran, and V. Bharghavan, "An integrated source transcoding and congestion control paradigm for video streaming in the internet," *IEEE Transactions on Multimedia*, vol. 3, no. 1, pp. 18–32, 2001.
- [19] V. M. Stanković, R. Hamzaoui, Y. Charfi, and Z. Xiong, "Real-Time unequal error protection algorithms for progressive image transmission," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 10, pp. 1526–1535, 2003.

- [20] R. Hamzaoui, V. Stanković, and Z. Xiong, "Optimized error protection of scalable image bit streams," *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 91–107, 2005.
- [21] S. Dumitrescu, X. Wu, and Z. Wang, "Efficient algorithm for globally optimal uneven erasure-protected packetization of scalable code streams," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME '06)*, vol. 2006, pp. 605–608, Toronto, ON, Canada, July 2006.
- [22] P. G. Sherwood and K. Zeger, "Progressive image coding for noisy channels," *IEEE Signal Processing Letters*, vol. 4, no. 7, pp. 189–191, 1997.
- [23] B. A. Banister, B. Belzer, and T. R. Fischer, "Robust image transmission using JPEG2000 and turbo-codes," *IEEE Signal Processing Letters*, vol. 9, no. 4, pp. 117–119, 2002.
- [24] V. Chande, N. Farvardin, and H. Jafarkhani, "Image communication over noisy channels with feedback," in *Proceedings of IEEE International Conference on Image Processing (ICIP '99)*, vol. 2, pp. 540–544, Kobe, Japan, October 1999.
- [25] V. Chande and N. Farvardin, "Progressive transmission of images over memoryless noisy channels," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 850–860, 2000.
- [26] S. Appadwedula, D. L. Jones, K. Ramchandran, and I. Konzentsev, "Joint source channel matching for a wireless communications link," in *Proceedings of the IEEE International Conference on Communications (ICC '98)*, vol. 1, pp. 482–486, Atlanta, Ga, USA, June 1998.
- [27] F. Etemadi, H. Yousefi'zadeh, and H. Jafarkhani, "Progressive bitstream transmission over tandem channels," in *Proceedings of IEEE International Conference on Image Processing (ICIP '05)*, vol. 1, pp. 765–768, Genova, Italy, September 2005.
- [28] E. Salemi, C. Desset, A. Dejonghe, J. Cornelis, and P. Schelkens, "A low-complexity methodology for unequal error protection of scalable images," in *Proceedings of IEEE Global Telecommunications Conference (GlobeCom '06)*, pp. 1–5, San Francisco, Calif, USA, November-December 2007.
- [29] E. Salemi, C. Desset, A. Dejonghe, J. Cornelis, and P. Schelkens, "Impact of source-independent modeling on unequal error protection for JPEG2000 images," in *Wavelet Applications in Industrial Processing IV*, vol. 6383 of *Proceedings of SPIE*, pp. 1–8, San Jose, Calif, USA, October 2006.
- [30] A. Nosratinia, J. Lu, and B. Aazhang, "Source-channel rate allocation for progressive transmission of images," *IEEE Transactions on Communications*, vol. 51, no. 2, pp. 186–196, 2003.
- [31] E. Salemi, C. Desset, J. Cornelis, and P. Schelkens, "Additive distortion modeling for unequal error protection of scalable multimedia content," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, vol. 2, pp. 269–272, Philadelphia, Pa, USA, March 2005.
- [32] USC-SIPI image database, available at <http://sipi.usc.edu/database>.
- [33] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, 1996.
- [34] S. D. Taubman, *JPEG2000: Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers, Norwell, Mass, USA, 2001.
- [35] M. Zhao and A. N. Akansu, "Optimization of dynamic UEP schemes for embedded image sources in noisy channels," in *Proceedings of IEEE International Conference on Image Processing (ICIP '00)*, vol. 1, pp. 383–386, Vancouver, BC, Canada, September 2000.