ADDITIVE DISTORTION MODELING FOR UNEQUAL ERROR PROTECTION OF SCALABLE MULTIMEDIA CONTENT

Eric Salemi[1,2], *Claude Desset*[1], *Jan Cornelis*[1,2], *Peter Schelkens*[1,2]

[1]IMEC, Kapeldreef 75 B-3001 Leuven, Belgium

[2]VUB Dept. ETRO, Pleinlaan 2 B-1050 Brussel, Belgium

Email: esalemi@imec.be
Phone: +32.16.28.89.39

ABSTRACT

Due to the improvement of compression technology, limited bandwidth channels can convey more data than before. However, since compressed streams are very sensitive to transmission errors, a joint optimization of the source-channel coding is required. Unequal Error Protection (UEP) is the most pragmatic way to do it, and exploits the embedded nature of bitstreams produced by scalable coders.

However, depending on the number of substreams and protection levels, optimization of UEP can be extremely complex. A large simplification comes from considering the contribution to the global distortion of each substream as independent from the others. This paper explores this distortion additivity assumption for JPEG2000, and shows that though the assumption does not fully hold, no visible consequence on UEP performance is introduced, while reducing significantly the complexity of the optimization process.

1. INTRODUCTION

A typical multimedia communication system can be decomposed in two fundamental steps. The source coding step aims at reducing the amount of data needed for the transmission while retaining maximal perceived quality at the output. In a second step, the channel coding protects the source coded data against channel impairments. By adding structured redundancy, the system is able to recover from errors that occur during the transmission.

Shannon's theory allows a separated design of the aforementioned steps to achieve error-free transmission, but only under the ideal assumption that we have an infinite computational power and data set. Though solutions have been developed to independently tackle source and channel impairments, joint optimization has recently gained increased interest, allowing global distribution of the available resources in a multimedia communication system [1].

Several techniques bring accurate control over the resource distribution. Scalability features, embedded in state-of-the-art source coders like JPEG2000 or SPIHT [2, 3], offer a fine quality control by simple selection of the amount of transmitted data. They enable graceful adaptation for wireless channels and naturally sort data based on its importance with respect to reconstruction quality. Secondly, channel coding techniques [4] provide smooth redundancy control in order to adjust the degree of protection applied to the data.

Combination of these techniques facilitated the developement of UEP strategies [5, 6, 7]. UEP conceptually exploits the significance and error sensitivity of the various parts of the transmitted data stream, and naturally selects a suitable protection level for each part. While UEP theoretically brings performance gain, we have little information on how to optimally assign the protection levels. Actually, most compression techniques result in the data streams highly sensitive to corruption. Moreover, the embedded nature of the scalable content introduces decoding dependencies among the substreams. Those effects are reflected in the error propagation behavior and harden the optimization of UEP algorithms.

Exhaustive search algorithms are optimal but they impose a significant complexity increase when the number of substreams or protection possibilities grows. Assuming that each substream of the scalable content has an independent contribution to the global distortion (i.e the contributions are additive) would significantly reduce the complexity of the final algorithm. While keeping an optimal global solution, it would allow to separately model each substream. Consequently to justify this simplification, the impact of substream dependencies on the distortion modeling and on the optimality of related UEP algorithms has to be studied. We carried out our investigations with the JPEG2000 source coder which is likely to become a prominent standard technology for future multimedia communication systems.

Section 2 summarizes the architecture of a JPEG2000 source coder and Section 3 presents the dependency structure of its internal data building blocks and the various tools, which control the impact of error propagation on the output distortion. Section 4 presents the additivity mismatch behavior of a corrupted JPEG2000 codestream without UEP related optimizations and Section 5 discusses the impact of additivity mismatch on a full-search UEP algorithm. Section 6 analyzes results and evaluates the validity of our assumption.

2. JPEG2000 STRUCTURE

JPEG2000 is a wavelet-based image compression standard using the EBCOT (Embedded Block Coding with Optimized Truncation) algorithm [2]. The encoding module transforms a still image into a hierarchical structure composed of multiple resolution levels, subbands, codeblocks, and chunks.

The discrete wavelet transform first decomposes each color component into several *resolution levels*, each containing a series of *subbands*. The coefficients in each wavelet subband are then quantized and divided into regular arrays of *precincts* and *codeblocks* for entropy coding.

Each codeblock is independently entropy-coded using the recursive probability interval subdivision of Elias coding [8]. Each entropy-coded codeblock is composed out of several coding passes (each bitplane taken out of a codeblock spawns 3 different coding passes), commonly referred to as *chunks*, and each chunk provides a variable quality contribution to the reconstructed image.

A post-compression rate-allocation Lagrangian-type algorithm subsequently selects chunks and packetizes them into *data packets*, which are assembled to form the final codestream. Data packets are the main building blocks of the JPEG2000 codestream. It is composed of two parts: a header and a body. The header indicates which chunks are included in this data packet and provides their local position. The body contains the actual data of the included chunks.

3. ERROR PROPAGATION MECHANISM

Through the rest of this document, the *substream* entity will reference the body of a data packet. Globally we want to focus on the statistical impact of the source data corruption only and hence do not consider corruption of the container itself (header information, markers, etc.). Though this assumption is in reality not valid, we except this to be mitigated by assigning a high-redundancy protection to the container as part of the UEP assignment. We prefer not to extensively broaden the scope of this document by analyzing the non-linear behavior of this type of corruption, but we definetely want to include it in our future work.

JPEG2000 embeds some basic error-resilience tools in its codestream [9]: (1) SOP (start of packet) markers allow partitioning and resynchronization at the packet level; (2) Entropy coding is done independently on each codeblock so that the distortion effect may not propagate into another codeblock and (3) segmentation symbols are used at the bitplane level to check the correctness of the decoding process (error detection tool).

The default decoding mode does not embed error-resilience tools at the chunk level, which may let errors propagate across bitplanes. Certain non-default modes help stopping the propagation by resetting the state of the arithmetic coder after each coding pass. Though these different modes allow to tune the corruption impact by trading off complexity against bandwidth consumption, we noticed little performance difference with default mode in all our simulations.

To summarize, JPEG2000 encoding process avoids that decoding errors propagates outside of the entropy-coded codeblock bounds, but the decoding dependency between its constitutive chunks may let errors propagate from one chunk to another. Consequently, a single bit error occuring in the most important chunk may affect the decoding of all subsequent chunks, and thus introduces errors in all corresponding bitplanes.

Thus, the error propagation due to early desynchronization in the entropy-coded codeblock causes errors in every bitplane. A second bit error in a lower chunk would have then a smaller contribution to the codeblock distortion, since decoding errors due to the first error are already generating distortion in the lower bitplanes. Eventually, the sum of the distortions generated by single errors is likely to be larger than the distortion caused by simultaneous bit errors, and causes a distortion additivity mismatch at the chunk level.

This effect is the first cause to distortion additivity mismatch between substream since chunks belonging to a single entropy-coded codeblock are distributed across substreams. It is further examined in Section 4 and Section 5, and supported by simulations.

A second cause to additivity mismatch, is the image-dependent pixel value distribution that introduces correlation between bit values across bitplanes. Although this effect is theoretically present, the use of an efficient data decorrelation stage like the wavelet transformation in JPEG2000 in combination with an entropy coder mitigates this effect.

Third, the inverse wavelet transform ideally sums up distortion effect originating from different substreams for any single pixel. But restrictions on the current decoders implementation may cause the computer arithmetic to overflow if summed distortion is larger than the anticipated bit-range.

4. MISMATCH EVALUATION

A simulation has been carried out that supports the assumptions we made so far on the additivity mismatch that may subsists after corruption of a JPEG2000 standard codestream.

We used the JPEG2000 Kakadu v4.0.3 source coder with the following options: SOP and EPH (end of packet) marker, lossless compression, 1 wavelet level (in addition to the base resolution level), 2 quality layers and 8x8 codeblocks. The transmitted image is a grayscale version of Lena with a 512x512 pixel resolution. The decoding process uses the resilient SOP marker mode. The total number of substreams in the encoded image is $S = 4$ (2 quality layers per resolution level). Uniform bit error rates (BERs) ranging from 10^{-5} 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 output distortion d^j (j stands for joint corruption). Then 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^i where $1 \leq s \leq S$ (i stands for individual). The indices i and j stand for joint or individual corruption methods. Figure 1 shows the additivity mismatch defined as $\alpha = 1 - (d^j / \sum_{s=1}^{S} d_s^i)$, which happens to be strictly positive. This confirms that the additivity based distortion estimation overestimates the real joint distortion. We observe a peak mismatch of approximately 12% on the MSE (i.e. less than 0.5 dB for the PSNR). However, the peak occurs in a high BER region, which will unlikely happen in an optimized communication system where BERs below 10^{-2} are typically observed. This translates on Figure 1 by a maximum MSE deviation of 4% (i.e less than 0.2 dB for PSNR) for BER close to 10^{-2}, and a rapidly decreasing mismatch as the BER goes to 0. Globally, it appears that the error propagation behavior is approximately additive.

UEP algorithms optimally match protection levels to the importance of each substream. By increasing the protection of important substreams we expect to lessen their large contribution to the distortion. Hence, we expect UEP to mitigate the masking effect which is one of the main cause for the additivity mismatch.

It is not clear if the additivity mismatch has a real impact on the optimality of the UEP distribution algorithm with regards to the reconstructed quality. Therefore the effect of the additivity mismatch on the quality of the reconstructed image quality has to be investigated.

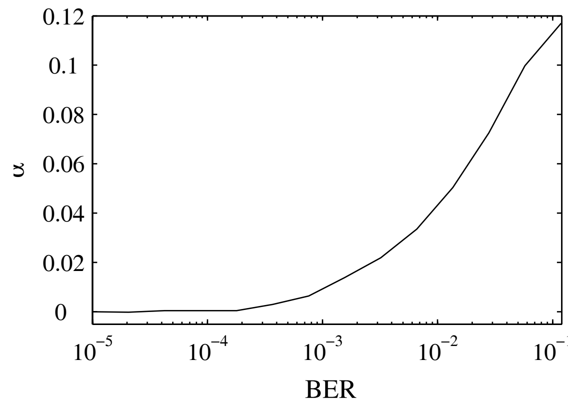

Fig. 1. Additivity Mismatch in Lena image

5. UEP SIMULATION

To fairly match the comparison made in Section 4, two UEP algorithms were implemented: a full-search (FS) and a full-search additive (FSA). For each channel signal-to-noise ratio E_s/N_0, where E_s is the uncoded symbol energy and N_0 is the noise power spectral density, we generate all possible protection level combinations, based on the number of substreams, the number of protection levels and a global bit-budget constraint that we fix at 2 bits per pixel (bpp). Note that each combination assigns to each substream a unique protection level. If needed we allow the algorithms to exactly fill the budget by protecting a part of the last assigned substream with a second lower protection level. Finally, for each combination we run 100 simulations and for each E_s/N_0 channel quality we pick up the combination that minimizes the output distortion.

The main differentiator between the two algorithms concerns the way they evaluate the output distortion for each protection level combination. FS does not use any distortion modeling. It straightly applies each combination to the substreams and outputs the average measured distortion. On the other hand, FSA uses the fact that substream distortions are additive. Consequently, it measures the average distortions of the individual substreams and sums them to estimate the output distortion. The main advantage of FSA is that for each channel quality, we only need to compute the individual average distortions *once* for each protection level, and sum their different combinations to obtain the global distortion. Denoting S the number of substreams and P the number of protection levels, we roughly bring down the complexity from an exponential P^S number of combinations (where an average over many simulations takes place) to a linear $P * S$, considering the computation of the P^S subsequent sums neglectable compared to simulations.

Figure 2 shows the PSNR performance of different UEP algorithms over a range of E_s/N_0. We use the same image and source coder parameters as in Section 4. BPSK symbols are used and the channel is modeled by an additive white Gaussian noise (AWGN) process. Two protection levels are available, a rate 1/2 convolutional encoder with generator polynomials (5,7) and a rate 4/5 punctured version of it. We additonally consider that a substream can be transmitted without any protection or discarded. As we deal with 4 substreams and 4 protection levels we subscript the corresponding algorithms $\mathrm{FS}_{4,4}$ and $\mathrm{FSA}_{4,4}$ on Figure 2.

For these two algorithms, the PSNR ranges from 13 dB to 36 dB. For E_s/N_0 below -3 dB, both algorithms do not transmit anything, so that the x-axis on Figure 2 is truncated accordingly. For E_s/N_0 above 11 dB, the PSNR seems to saturate. Indeed, bit-rate is limited to 2 bpp while the bitrate of the encoded image is close to 4.5 bpp. Hence, for an asymptotic error-free channel, the quality is limited by the lossy compression. Globally, both algorithms reach similar transmission performances, which is confirmed by the fact that their respective protection allocations are strictly identical.

Hence, the $\mathrm{FSA}_{4,4}$ algorithm performs equivalently compared to the exhaustive $\mathrm{FS}_{4,4}$ algorithm. Moreover, the UEP algorithm complexity is significantly reduced when assuming distortion additivity. Indeed, each substream can be modeled independently and an optimal distortion estimation for the complete codestream is given by simply calculating a sum.This proves that the additivity assumption brings a huge simplification to the complexity of UEP algorithms.

We can discern three PSNR thresholds occuring around channel quality -1 dB, 3 dB and 9 dB which correspond to the 3 transitions that we can make out of the 4 protection levels. Indeed, both algorithms always keep a nearly constant protection level throughout the bitstream between 2 consecutive thresholds. However, for channel qualities around those thresholds, both algorithms optimally use different protection levels with seamless transitions. The restricted number of substreams and number of protection levels used for this simulation result in an undesirable, but yet optimal, staircase effect. An additional $\mathrm{FSA}_{16,4}$ simulation with increased number of substreams is shown on Figure 2, exhibiting smoother transitions and a significantly reduced staircase effect.

6. CONCLUSION

This paper shows that the contributions to the global distortion of the substreams embedded in a JPEG2000 scalable codestream are relatively independent. It has been proven that for a standard JPEG2000 codestream, the additivity assumption significantly simplifies the computational complexity of the UEP algorithm without impact on the performance.

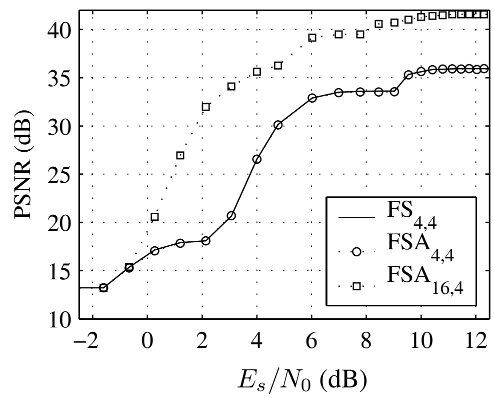

Fig. 2. FS vs. FSA performance for Lena at 2 bpp

7. REFERENCES

[1] Gene Cheung and Avideh Zakhor, "Bit allocation for joint source/channel coding of scalable video," *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 340–356, Mar. 2000.

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

[3] Ali Said and William A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierachical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, June 1996.

[4] Joachim Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Transactions on Computers*, vol. 36, no. 4, pp. 389–400, Apr. 1988.

[5] A. Natu and David Taubman, "Unequal protection of JEPG2000 code-streams in wireless channels," *Global Communication Conference, GLOBECOM*, vol. 1, pp. 534–538, Nov. 2003.

[6] 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, Feb. 2003.

[7] Vladimir M. Stanković, Raouf Hamzaoui, Youssef Charfi, and Zixiang 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, Dec. 2003.

[8] Charilaos Christopoulos, Athanassios Skodras, and Touradj Ebrahimi, "The JPEG2000 still image coding system: An overview," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 1103–1127, Nov. 2000.

[9] Ali Bilgin, Zhenyu Wu, and Michael W. Marcellin, "Decompression of corrupt JPEG2000 codestreams," in *Proceedings of Data Compression Conference*, Mar. 2003, pp. 123–132.