JOINT SOURCE-CHANNEL DECODING OF JPEG2000 IMAGES WITH UNEQUAL LOSS PROTECTION

Sohail Bahmani, Ivan V. Bajić, and Atousa HajShirMohammadi

School of Engineering Science, Simon Fraser University, Burnaby, BC, V5A 1S6, Canada

ABSTRACT

This paper presents a method for joint decoding of JPEG2000 bitstreams and Reed-Solomon codes in the context of unequal loss protection. When the Reed-Solomon decoder is unable to retrieve the erased source symbols, the proposed joint decoder searches through the set of possible erased source symbols, making use of error resilience features of JPEG2000 to retrieve correct symbols. The joint decoder can be used as an add-on module to some of the existing schemes for unequal loss protection, and can improve the PSNR of decoded images by over 10 dB in some cases.

Index Terms— Unequal loss protection, JPEG2000, Reed-Solomon codes, joint source-channel decoding.

1. INTRODUCTION

Motivated by recent developments in scalable image and video coding, and the emergence of scalable coding standards such as JPEG2000 [1] and H.264/SVC [2], researchers have been investigating various methods for robust transmission of scalable bitstreams. The progressiveness of scalable bitsreams makes them different from plain data bitstreams, and more prone to corruption by errors and erasures. In a scalable bitstream, the effect of the corruption or erasure of different bits on the decoded signal quality is different. Consequently, Unequal Error Protection (UEP) and Unequal Loss Protection (ULP) schemes are thought to be very appropriate for protecting scalable bitstreams against errors and erasures, respectively.

Our focus in this paper is the transmission of JPEG2000 bitstreams over packet-based networks, hence we will focus on ULP schemes for this scenario. Several methods, e.g., [3], [4], [5], have been developed for ULP of scalable image bitstreams by erasure correction codes. In these schemes, each segment of the scalable bitstream is protected by an appropriate amount of redundancy, chosen according to the segment's "importance." For a given channel model, redundancy allocation is performed to maximize the expected quality of the received image. Several optimization techniques were proposed in [3]-[5] for solving this optimization problem, including hill climbing and local search, each with different complexity and performance.

In this paper, we develop a joint source-channel decoding scheme for JPEG2000 bitstreams with ULP. The proposed decoding scheme is complementary to the ULP methods mentioned above, and can be used with each of them to improve the quality of decoded images. The proposed scheme takes advantage of Error Resilience (ER) features of the arithmetic coder employed in JPEG2000. These features have been used in [6] for joint source-channel decoding of JPEG2000 and LDPC codes over error-prone Additive White Gaussian Noise (AWGN) channels. Here, we demonstrate how these ER features can also be used for JPEG2000 bitstream transmission over erasure channels.

The paper is organized as follows: In sections 2 and 3, we provide further details about the ULP scheme and ER features of JPEG2000 used in this work. In Section 4 we describe the proposed joint source-channel decoding strategy. Results and conclusions are presented in Section 5.

2. UNEQUAL LOSS PROTECTION

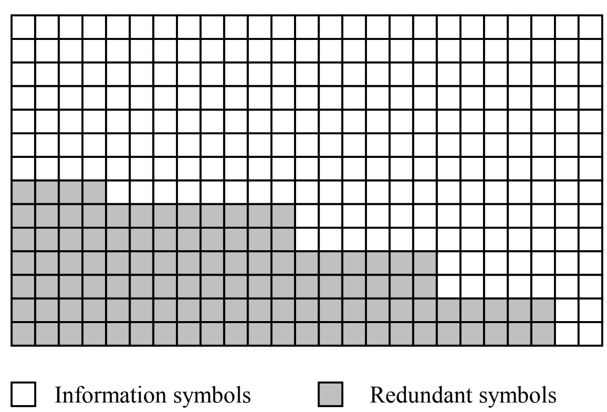

Fig. 1. A typical ULP structure.

This work was supported in part by the SFU Start-up Grant and NSERC Grants RGPIN 327249 and EQPEQ 330976.

A typical ULP structure is illustrated in Fig. 1, where white blocks indicate information symbols of the source bitstream, and gray blocks indicate redundancy symbols. Each column represents a codeword of a systematic erasure-correction code (Reed-Solomon (RS) code in this work), while each row represents a packet to be transmitted through the packet-based network. Information symbols are filled column-by-column and after generating redundant symbols, they are transmitted row-by-row.

A (N, m, k) RS code has a total of N symbols in the codeword, k of which are information symbols; m is the number of bits that make up each symbol, hence $N \le 2^m$. The code with these parameters is able to correct up to $N - k$ symbol erasures. In the ULP scheme in Fig. 1, N is constant while k varies according to the desired protection level for the corresponding information symbols. Let k_i be the number of information symbols in the i-th column. We refer to the set of columns with the same protection level (i.e., same k_i) as a *segment*. For progressive bitstreams, protection level decreases towards the end of the bitstream, so k_i increases (i.e., $N - k_i$ decreases) as we move from the leftmost column to the rightmost column in Fig. 1.

Suppose L packets get lost during transmission. The set of columns where erased information symbols can be recovered by RS decoding is $C = \{i : L \le N - k_i\}$, and the corresponding number of decodable source bits is

$$R = \sum_{i \in C} m \times k_i \ . \tag{1}$$

This number can be increased slightly by decoding the initial portion of the first column in which $L > N - k_i$, up to the location of the first erasure, at which point decoding stops. Our goal here is to develop a method for decoding beyond this point. To achieve this goal, we exploit ER features of JPEG2000 described in the next section.

3. ERROR RESILIENCE IN JPEG2000

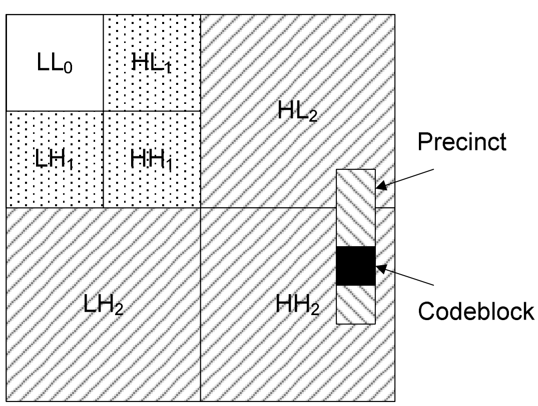

Fig. 2. Tile-component in JPEG2000 image coding.

To encode the raw image, JPEG2000 first divides it into disjoint rectangular *tiles*. The subband/wavelet transform is applied to each tile-component to generate subbands, which are then divided into rectangular-shaped *precincts*, and further divided into square-shaped *codeblocks*, as shown in Fig. 2. Each bitplane of a codeblock is then encoded by an arithmetic encoder in three *coding passes*. This provides a progressive bitstream for each of the codeblocks. Coding passes are then interleaved to create the scalable JPEG2000 bitstream [1]. To satisfy the requirements of our joint source-channel decoding, we used a single tile and a single quality layer in JPEG2000 encoding. This way, the bits from any given codeblock appear as a contiguous segment in the JPEG2000 bitstream. In fact, with the default ordering of scalability options, using one quality layer will generate a resolution-scalable bitstream [1].

The arithmetic coder in JPEG2000 includes several ER features which make the decoder able to detect bit errors and avoid error propagation. Of particular interest to us are the so-called RESTART mode and ERTERM mode. When the RESTART mode is used, the arithmetic encoder is restarted at the beginning of each coding pass. Further, when the ERTERM mode is switched on, the encoder uses a predictable termination policy for each arithmetic codeword segment. Bit errors disrupt synchronization between arithmetic encoder and decoder, typically resulting in erroneous termination of arithmetic codewords which can be detected in the ERTERM mode. It has been reported in [6] that when RESTART and ERTERM modes are used, bit errors can be detected within three coding passes in over 99% of the cases. Further details of ER features for JPEG2000 can be found in [1] and [7].

4. JOINT SOURCE-CHANNEL DECODING FOR ULP

ER features of JPEG2000 have been used in [6] for joint source-channel decoding with LDPC codes. There, ER features are exploited to update the soft information of the LDPC soft decoder and hence improve the quality of decoded images. In the following paragraphs, we explain how we used these features for joint decoding with RS codes.

Let L be the number of lost packets in the ULP structure in Fig. 1. As explained above, RS decoder can reconstruct erased source symbols in all columns in which $L \le N - k_i$ and the source bitstream will be decodable up to the rate R given in eq. (1). We wish to decode beyond this point. Let

$$k^{CS} = \min_{L > N - k_i} k_i \tag{2}$$

be the number of information symbols in those columns whose protection level is just below the level necessary to recover L erasures. We define the *critical segment* (*CS*) as the set of all such columns: $CS = \{i : k_i = k^{CS}\}$. The difference between the protection level in CS, and protection level that would be sufficient to reconstruct L erasures is given by $d = L - (N - k^{CS})$.

Now, if we guess d erased symbols in each column in CS, the remaining number of erased symbols will be $N - k^{CS}$, so the RS decoder will be able to fill in the remaining erased symbols. Since each symbol has m bits, there are $2^{d \times m}$ possible choices for the d symbols. But how can we tell whether the guess was correct? Each particular choice of the d symbols would lead to a valid RS codeword, but not necessarily to a valid JPEG2000 bitstream. This is where ER features of JPEG2000 can help us identify the correct guess: incorrect guesses for the d symbols will lead to bit errors in the JPEG2000 bitstream, which can be captured by the JPEG2000 decoder. A simplified version of the proposed joint source-channel decoding algorithm can be summarized with the following pseudo-code.

```
identify CS;
for each column i in CS
        G:     guess d erased symbols;
        RS:    let RS decoder fill the
               remaining erased symbols;
        J2K:   decode the information bits
               from column i;
               if error, go back to step G;
               else i++;
end
```

Upon the execution of this algorithm, information bits from CS are decoded, so the total number of decoded bits is

$$R^{+} = \sum_{i \in C \cup CS} m \times k_i \,, \tag{3}$$

which is greater than R from eq. (1) whenever CS exists (i.e., whenever it is non-empty). If we can afford to guess more than d symbols, the algorithm can be extended in a straightforward manner to decode additional bits beyond CS. In practice, JPEG2000 decoder cannot always detect an error caused by an incorrect guess in the same column – it may happen that the error is reported several column down the line. This usually happens when the coding pass occupies two or more columns. In those cases, we have to go back several columns to refine the guess that caused the error, and then resume decoding.

The improvement in decoded image quality brought by decoding additional bits from CS depends on many factors, including the image itself, the algorithm employed for ULP optimization, and the number of lost packets. On the other hand, the penalty paid for decoding additional bits is the increased complexity due to iterative guessing of erased symbols in CS. In the next section, we quantify both the improvements and the increase in decoding time for several standard images.

5. RESULTS

In our simulations, we used 8-bit symbols, i.e. m = 8. This leads to RS codes of the type (256, 8, k). To keep the complexity reasonable, we have limited our simulations in the following ways. We have only considered the cases where d = 1, in order to limit the number of possible guesses in each column to 2^8 = 256. Moreover, when all possible guesses for a particular column lead to an error (indicating that a guess in one of the preceding columns was wrong), we only let the decoder go back up to two columns to correct its guesses. As mentioned previously, these cases usually happen when a coding pass occupies more than one column. All tests were carried out with the Kakadu implementation of JPEG2000 codec [1] and Phil Karn's RS codec [8]. ULP assignment was performed using the optimization method and channel model from [5], for 10%, 20% and 30% average channel loss rate. Please note that due to the fact that our joint decoding scheme can usually decode more bits than standard decoding, the optimization method from [5] is not necessarily optimal in this context, but we used it for convenience. For further details on the channel model and ULP optimization, the readers are referred to [5].

Experiments were carried out on three standard 512×512 grayscale images, *Lena*, *Barbara and Gold Hill* Each image was encoded in to 255 packets of length 100 bytes, which corresponds to the total rate budget of roughly 0.78 bits per pixel. Table 1 shows the average improvement in the Peak Signal-to-Noise Ratio ($\overline{\Delta PSNR}$) in dB brought by the proposed joint source-channel decoding algorithm. For each image and each ULP profile (obtained by the method from [5] with the given channel loss rate), we test joint source-channel decoding on the last three segments (*CS* #1, *CS* #2, and *CS* #3). In each case, we fix the number of erasures, generate 100 erasure patterns, and perform decoding for each erasure pattern. The results represent averages taken over these 100 different erasure patterns. We also report the width (number of columns) of each segment in Table 1. In three cases (for *Lena*, *Barbara and Gold Hill* at 30% loss rate), the optimization method from [5] converged on a ULP assignment with only two segments, so the results for the third segment could not be obtained.

Table 1

Average PSNR improvement (in dB) and column width for the last three segments.

Image	Loss rate	*CS* #1		*CS* #2		*CS* #3	
		$\overline{\Delta PSNR}$	Width	$\overline{\Delta PSNR}$	Width	$\overline{\Delta PSNR}$	Width
Lena	10%	+1.6	9	+7.3	57	+4.5	23
	20%	+11.6	86	+2.4	8	0.0	1
	30%	+6.5	60	+14.3	40	-	-
Barbara	10%	+10.9	89	0.0	1	+0.5	2
	20%	+7.3	87	+0.7	5	+1.0	3
	30%	+2.2	20	+14.5	80	-	-
Gold Hill	*10%*	+7.0	85	+0.3	5	+1.3	4
	20%	+1.0	83	+0.3	7	+1.2	6
	30%	+0.2	6	+12.4	94	-	-

As expected, the improvement in decoded image quality depends on the number of bits in the *CS*, which in turn depends on the number of columns in the *CS*. For *CS*'s with only one column, the improvement is marginal (less than 0.05 dB). For *CS*'s with a large number of columns (e.g. 40 or more), the improvements can be over 14 dB. Visual quality improvement is illustrated in Fig. 3.

Fig. 3. *Barbara* image after standard ULP decoding, PSNR = 22.8 dB (top); and after additional joint source-channel decoding of the first CS, PSNR = 29.6 dB (bottom), for 20% loss rate.

Table 2, shows the average increase in decoding time ($\overline{\Delta T}$, in seconds) due to joint source-channel decoding, (JSCD), followed by the average decoding time without the use of our JSCD method (*T*, in brackets). The simulations were run on a desktop PC with an Intel Core 2 Duo 2.13 GHz CPU with 2GB of RAM. Note that the executables were compiled and run in "Debug" mode, which is not optimized for speed. Hence, the reported times should be taken as an upper bound on decoding time increase. The results indicate that decoding time generally increases with the width of the segment, but not as clearly as the corresponding quality improvement.

Since the number of guesses (and hence the decoding complexity) depends on the size of RS symbols among other things, it might be possible to optimize the symbol size in order to reduce the complexity. This is a possible topic for future research.

Table 2

Average increase $\overline{\Delta T}$ in decoding time (in seconds) when JSCD is used; decoding time *T* without JSCD is shown in brackets.

Image	Loss rate	*CS* #1	*CS* #2	*CS* #3
		$\overline{\Delta T}$ (*T*)	$\overline{\Delta T}$ (*T*)	$\overline{\Delta T}$ (*T*)
Lena	10%	1.49 (0.17)	8.82 (0.10)	7.45 (0.07)
	20%	221.23 (0.09)	2.46 (0.07)	0.74 (0.07)
	30%	206.22 (0.20)	338.17 (0.06)	-
Barbara	10%	69.36 (0.08)	0.22 (0.07)	0.39 (0.07)
	20%	352.70 (0.09)	1.51 (0.08)	0.81 (0.07)
	30%	341.87 (0.35)	62.88 (0.06)	-
Gold Hill	*10%*	220.22 (0.08)	0.93 (0.07)	0.56 (0.07)
	20%	789.71 (0.16)	2.29 (0.08)	2.00 (0.07)
	30%	2.91 (0.39)	1537.1 (0.06)	-

ACKNOWLEDGEMENT

The authors would like to thank Prof. Michael W. Marcellin for his comments on the JPEG2000 bitstream structure.

REFERENCES

[1] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*, Kluwer, Norwell, MA, 2001.

[2] H.-C. Huang, W.-H. Peng, T. Chiang, and H.-M. Hang, "Advances in the scalable amendment of H.264/AVC," *IEEE Commun. Mag.*, vol. 45, no. 1, pp. 68-76, Jan. 2007.

[3] R. Puri and K. Ramachandran, "Multipe description source coding using forward error correction codes," *Proc. Asilomar*, vol. 1, pp. 342-346, Oct. 1999.

[4] A. E. Mohr, E. A. Riskin, and R. E. Ladner, "Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction," *IEEE Trans. Select. Areas Commun.*, vol. 18, no. 6, pp. 819-828, Jun. 2000.

[5] V. M. Stanković, R. Hamzaoui, and Z. Xiong, "Real-time error protection of embedded codes for packet erasure and fading channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 8, pp. 1064-1072, Aug. 2004.

[6] L. Pu, Z. Wu, A. Bilgin, M. W. Marcellin, and B. Vasic, "LDPC-based iterative joint source-channel decoding for JPEG2000," *IEEE Trans. Image Processing*, vol. 16, no. 2, pp. 577-581, Feb. 2007.

[7] A. Bilgin, Z. Wu, and M. W. Marcellin, "Decompression of corrupt JPEG2000 codestreams," *Proc.DCC*, pp.123-132, Mar. 2003.

[8] P. Karn, *C++ class library for Galois field arithmetic and algebra, with RS encoder/decoder*, [Online] Available: http://www.ka9q.net/code/fec/fec-3.0.1.tar.bz2