PAPER *Special Section on Selected Papers from the 16th Workshop on Circuits and Systems in Karuizawa*

An Efficient Error Correction Scheme Allowing Backward Compatibility with JPEG2000 Codestream

Masayuki KUROSAKI[†a)], ***Student Member and*** **Hitoshi KIYA**[†b)], ***Member***

SUMMARY A scheme of error correction for JPEG2000 codestream is proposed in this paper. The scheme uses a forward error correction code (FEC) and a data hiding technique. The headers and the higher quality layers of the codestream are coded using FEC codes. Then the parity data are separated from the FEC-coded data and hidden in the JPEG2000 codestream. The hidden data are used for error correction at the decoder. Several error correction codes with different strength are selected for the main header, the tile-part headers, the packet headers, and the bodies. The codestream generated by the proposed scheme has backward compatibility with a standard JPEG2000 codestream. Thus, it can be decoded with a general decoder. Simulation results demonstrated the effectiveness of the proposed scheme.

key words: *JPEG2000, error correction, FEC, data hiding*

1. Introduction

Multimedia communication through wireless networks and the Internet is becoming increasingly important. Since these networks transmit data at high to low bit error rates (BERs), some portions of the transmitted data may not be decoded. This adversely affects the quality of the decoded images. In communication systems, an automatic repeat request (ARQ) and a forward error correction (FEC) code are generally used to protect the transmitted bit stream. These schemes however require special protocols to transmit data and cannot achieve the ideal error-free transmission.

Handling these problems requires image coding schemes that take transmission errors into account. JPEG2000 part 1 [1] still-image coding and JPEG2000 part 3 (motion JPEG2000) [2] and MPEG-4 [3] video coding, for example, consider such errors [4]–[7]. They contain error resilience tools (ERT) to detect and localize the errors and also to resynchronize the decoding process. These tools can thus minimize the effect of errors on image quality. However, standard error resilience tools cannot correct error data. Consequently, additional schemes that can recover error data during decoding are required.

Various schemes of correcting and concealing errors have been proposed [8]–[19]. Most conventional schemes work only for MPEG and are not applicable to the JPEG2000 standard [8]–[13]. Several schemes for JPEG2000 have been considered in [14]–[19]. The ones in [14] and [15] are error protection schemes applied at physical and application levels, respectively. Although these schemes can correct errors in the JPEG2000 codestream, special decoders are needed to decode the coded data. On the other hand, other authors offered a concealment method that based on interpolating the wavelet coefficients [16] and jointly exploiting the intra- and inter- band correlation [17]. However, all the schemes described above cannot perfectly correct any errors. In addition, most conventional schemes assume that the headers are intact during transmission process, which is not always the case in practical applications.

In this paper, we propose a new scheme of error correction for JPEG2000 and motion JPEG2000 that provides backward compatibility to the standard decoders. The proposed scheme uses FEC codes, a data hiding technique, and the JPEG2000 layer structure. The main, tile-part, and packet headers and the more significant data in the higher quality layers of the JPEG2000 codestream are protected using the FEC codes, and the resulting parity data are separated from the FEC-coded data and then hidden in the JPEG2000 codestream. At the decoder, the hidden data is used for error correction. Several locations in which data is hidden exist in the JPEG2000 codestream. Here, the last layer of the codestream is considered due to some advantages.

The proposed method can have backward compatibility with the standard JPEG2000 by using a data hiding technique as well as described in the previous works [18], [19]. The codestream produced by the proposed scheme therefore can be decoded by a general decoder. In addition, since the proposed scheme separates the parity data from the FEC-coded data, the proposed scheme is more efficient in correcting errors under the same hidden data size than in [18] and [19].

The organization of this paper is as follows. Section 2 reviews the JPEG2000 encoder. In Sect. 3, the error correction for JPEG2000 codestream is proposed. Section 4 shows the results of some simulations of the proposed scheme. Finally, some concluding remarks are made in Sect. 5.

2. JPEG2000 Encoder

In this section, we briefly describe the JPEG2000 encoding procedure and packets, and the error resilience tools.

Manuscript received June 30, 2003.
Manuscript revised October 3, 2003.
Final manuscript received November 28, 2003.

†The authors are with the Department of Electrical Engineering, Tokyo Metropolitan University, Hachioji-shi, 192–0397 Japan.
a) E-mail: kurosaki@isys.eei.metro-u.ac.jp
b) E-mail: kiya@eei.metro-u.ac.jp

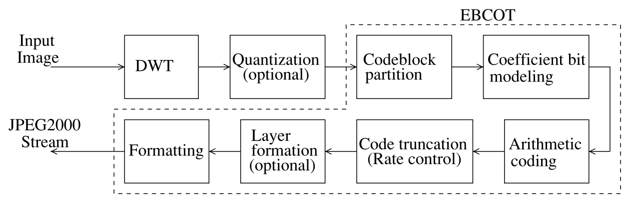


Fig. 1 JPEG2000 encoder.

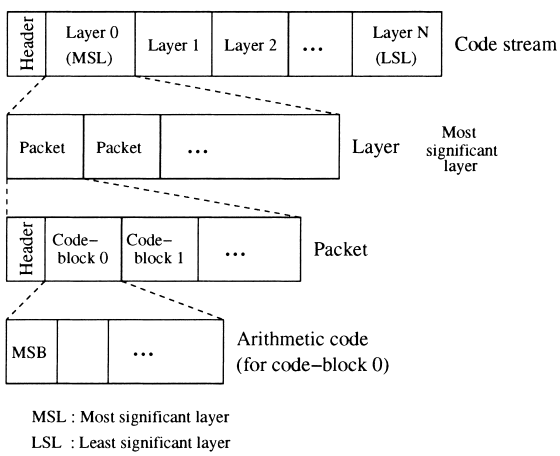


Fig. 2 Structure of JPEG2000 codestream.

2.1 Encoding Procedure

A block diagram of a JPEG2000 encoder is shown in Fig. 1. The input image is decomposed into its sub-bands by a discrete wavelet transform, and the wavelet coefficients of each sub-band are quantized. The quantized coefficients are coded using the EBCOT (embedded block coding with optimized truncation of the embedded bit streams) algorithm [20]. In this algorithm, each sub-band is partitioned into rectangular blocks (for example 64×64) called "code-blocks," which form the independent input to the coefficient bit modeling based on bit-plane and arithmetic coding. A layer structure is formed for scalability of the bit stream; the output codestreams are truncated so that they can achieve the target bit rate. Finally, headers are attached to create the JPEG2000 codestreams.

2.2 Layer Structure and Packets

The structure of a JPEG2000 codestream is shown in Fig. 2. The codestream is arranged in a layer structure according to its contribution to image quality. The more significant bit-planes of the code-blocks are included in the higher quality layers, and the less significant bit-planes are successively placed in the lower quality layers. This enables the decoder to decode a codestream progressively in terms of image quality. In a general JPEG2000 encoder, the number of layers and the length of each layer can be assigned arbitrarily.

Each layer is divided into multiple units called packets. A packet consists of a body and its header information. Each body has code-block information of the resolution level corresponding to each layer. Since the size of a packet is determined automatically, we cannot assign an arbitrary size.

Table 1 Error resilience tools in standard JPEG2000 coding [1].

Type of tool	Name
Entropy coding level	· Code-blocks · Termination of the arithmetic coder for each pass · Reset of contexts after each coding pass · Selective arithmetic coding bypass · Segmentation symbols
Packet level	· Short packet format · Packet with resynchronization marker

2.3 Error Resilience Tools

The tools available for error resilience in standard JPEG2000 coding are shown in Table 1. Error resilience is achieved at the entropy coding and packet levels. At the entropy coding level, to localize random and burst errors, the arithmetic coder can be terminated after each coding pass. The context can then be reset after each pass, enabling the decoder to continue the decoding process if errors are detected. Similarly, at the packet level, the decoder can detect and localize errors in the packets. However, data lost as a result of errors cannot be recovered using the standard error resilience tools. In particular, when significant data in lower layers and packets are lost, there is significant image deterioration.

3. Proposed Scheme

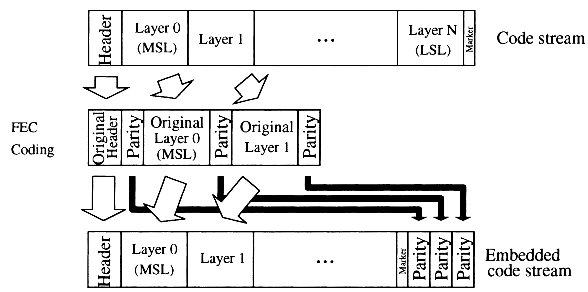
In the proposed method, the headers and some important data are FEC-coded. Then, the parity data is separated and hidden in the codestream. The main idea of the proposed scheme is to separate the parity data and to hide it in the JPEG2000 codestream. Since the hidden data is only the parity data, it enables us to protect a lager number of data and to use more powerful FEC codes. Therefore, the proposed method is more efficient than the ones in the previous works. In addition, the reason for hiding the parity data in the JPEG2000 code stream is to provide backward compatibility with the standard JPEG2000 codestream. In this section, we explain the scheme of hiding parity symbols in the JPEG2000 codestream, and the error correcting scheme. There are three schemes of hiding data in the codestream with backward compatibility, namely:

(a) Inserting the data on the comment area in the codestream.
(b) Replacing the least significant layer (LSL) of the codestream with the data.
(c) Attaching the data after the codestream.

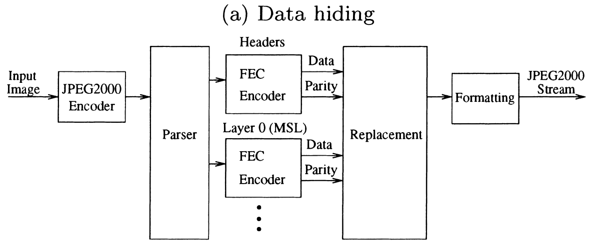
Some features (No degradation, simplicity of the process, and no additional bandwidth) of these schemes are summarized in Table 2.

Table 2 Features of data hiding schemes.

Scheme	No image degradation	Simplicity of process	No additional bandwidth
(a)	○	○	×
(b)	×	○	○
(c)	○	×	×



(a) Data hiding



(b) Extended JPEG2000 encoder

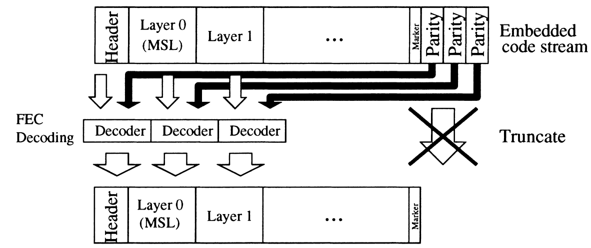
Fig. 3 Proposed JPEG2000 encoder.

The scheme (a) inserts the desired data in the comment (COM) area of the main and tile-part headers in the codestream. Several comment areas can be defined in one JPEG2000 codestream. The total size of the resulting codestream is larger than that of an original JPEG2000 codestream. In the scheme (b), the last layer is replaced with the desired data. Thus additional bandwidth is not required for transmission, although this scheme introduces a slight image degradation. In the JPEG2000 part 1 [1] and part 3 [2], the file formats are constructed of boxes that contain the codestream and/or the meta data. Therefore, if the size of the codestream is increased, modifying the header data in the boxes is required. The scheme (c) attaches the desired data after the end of the original JPEG2000 codestream. The resulting size of the codestream therefore increases to a size of the attached data. Moreover, this scheme requires further modification on the headers, hence, the process is more complex.

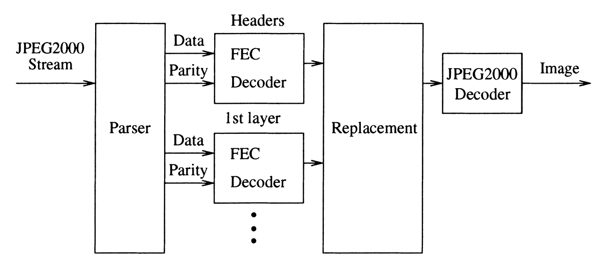
As an example, the scheme (b) will be explained in detail.

3.1 Data Hiding

Figure 3 illustrates the data hiding process and its implementation in an extended JPEG2000 encoder. First, layers, each with a specified data size, are generated using the standard JPEG2000 encoder. Then, the headers and the selected layers are FEC-coded. Next, the parity data are separated



(a) Error correction process



(b) Extended JPEG2000 decoder.

Fig. 4 Proposed JPEG2000 decoder.

from the FEC-coded data in the parsed codestream. Finally, the LSL is replaced by the parity data (see Fig. 3(a)). It is worth noting that any FEC codes can be used along with this scheme. Thus, for a higher error rate, a more powerful code can be chosen for error correction. In addition, various strength of the FEC code can be differently assigned for the main, tile-part, and packet headers, and the bodies.

The characteristics of the generated codestream are as follows.

- The length of the codestream with the hidden data is the same as that of the original codestream.
- Since the codestream has the same data structure as the standard JPEG2000 codestream, the codestream has backward compatibility with a standard JPEG2000 codestream, the codestream does not stop the decoding process at a general decoder. In additon, the standard JPEG2000 error resilience tools can also be used.
- The LSL, which is replaced by the parity data, can be truncated to inhibit image deterioration caused by data hiding.

3.2 Error Correction

Figure 4 illustrates the error correction process and its implementation in an extended JPEG2000 decoder. First, the erroneous headers and higher quality layers are recovered using the hidden parity data; the errored headers and high quality layers are then replaced with the corrected headers and layers, respectively. Finally, to inhibit image deterioration caused by data hiding, the LSL is truncated.

In this scheme, the FEC code can detect and correct errors. That is, errors in the headers and the higher quality layers are detected and corrected independently of the error resilience tools defined in the JPEG2000 standard. The proposed scheme can be combined with the standard error resilience tools and other error correction schemes.

It is worth pointing out that all the proposed error correction with the aforementioned hiding schemes provides the same capability with that using the hiding scheme (b).

4. Simulation

We evaluated the performance of the proposed scheme using some simulations. In this section, we applied the proposed error correction scheme to both the still-image and the video sequence.

4.1 Simulation Conditions

The test image and sequence are “Lena” (256 levels gray-scale, 512 × 512 pixels, 262,144 bytes) and the video sequence “Mobile & calendar” (256 levels gray-scale, 720 × 576 pixels, 40 frames), respectively. Image coding was done using five-level decomposition/composition based on two-channel filter banks with a Daubechies 9/7 bi-orthogonal wavelet filter running under JPEG2000 verification model 8.6 [21]. The target bit rates were 1.0 and 0.2 bits/pixel (bpp). The codestream consisted of 20 layers (layers 0-19). The errors were generated randomly and independently at rates of 10^{-1} to 10^{-7}.

Two different Reed-Solomon (RS) codes [22], RS(15, 13) with a one-symbol correctable error and RS(15,11) with a two-symbol correctable error, were used for error-correction coding to protect both of the headers and some important layers by using a FEC code, RS(15,13) or RS(1511). Considering the size of the lowest layer, as previously investigated in [18], the numbers of layers that can be corrected in the proposed scheme are seven (layer 0 to layer 6) and three (layer 0 to layer 2), respectively. In the propsed method, the LSL size is set to be larger than the hidden data length. The remainder of the LSL is padded by zero values. For the simulation, the headers are transmitted with error-prone.

Image quality was measured in terms of the peak signal-to-noise ratio (PSNR) between the original and the decoded image: the higher the PSNR, the better the image quality.

PSNR: 40.29 [dB]
(a) W/o hidden data.

PSNR: 39.93 [dB]
(b) With hidden data.

Fig. 5 Image deterioration caused by data hiding.

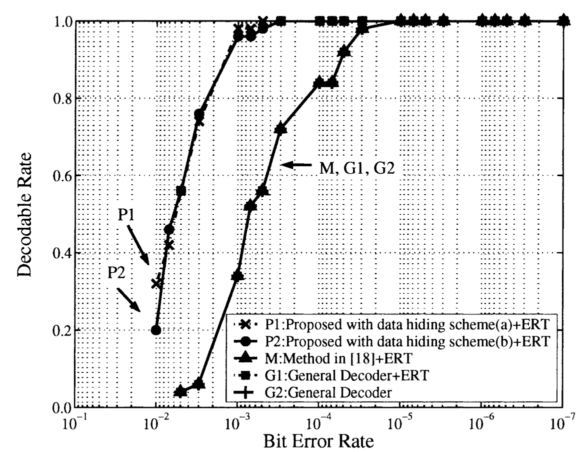
4.2 Image Deterioration Caused by Data Hiding

Figure 5 shows the images decoded from the original JPEG2000 codestream without and with 1890 bytes of hidden parity data. The PSNR values were 40.29 [dB] and 39.93 [dB], respectively (see Fig. 5). The hidden data was truncated from the codestream during decoding. (A general JPEG2000 decoder can truncate hidden data using marker codes.) As these results show, the proposed scheme can hide about 1890 bytes of data with very slight image deterioration (about 0.36 dB). This means that the image deterioration caused by data hiding can be neglected.

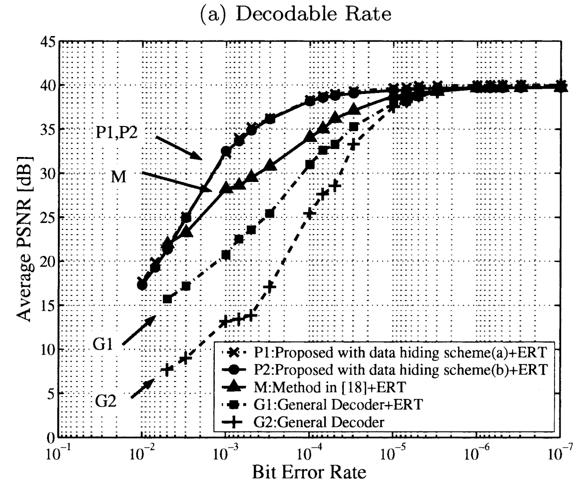
Under different hidden data length, the proposed scheme effectively inhibited image degradation due to data hiding [23].

4.3 Quality of Error Correction

(A) Quality of error correction for still-image
The simulation results show the decodable rates and the average PSNR values that were obtained by taking an ensem-

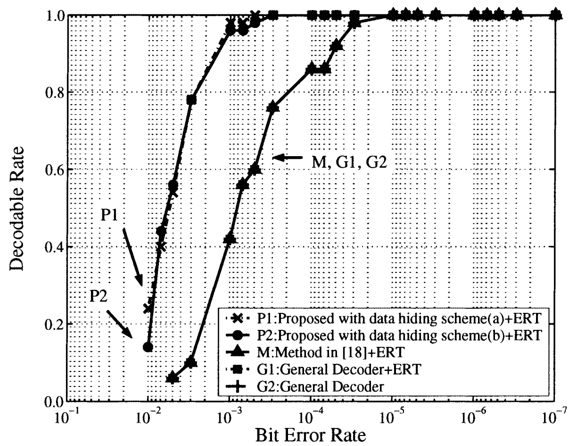


(a) Decodable Rate

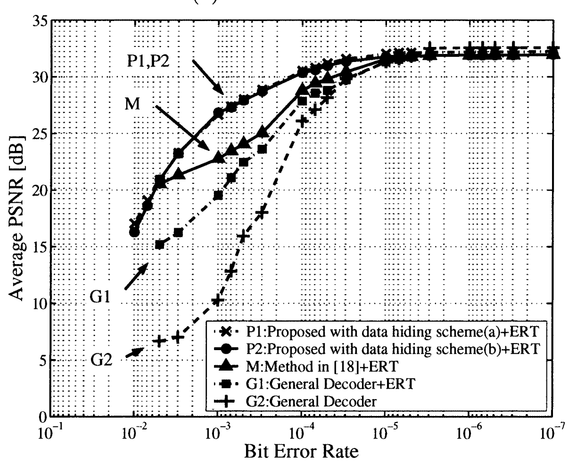


(b) Average PSNR

Fig. 6 Quality of proposed error correction with RS(15,13) for a random error. (target bit rate: 1.0 bpp)



(a) Decodable Rate



(b) Average PSNR

Fig. 7 Quality of proposed error correction with RS(15,13) for a random error. (target bit rate: 0.2 bpp)

ble average over 50 independent trials for the same BER condition.

Figures 6 and 7 show the values for images decoded from erroneous codestreams that are encoded by the proposed encoder at target bit rates of 1.0 bpp and 0.2 bpp, respectively. The decodable rate of 1.0 means that the codestreams were decoded by the decoder 50 times over the 50 independent traials. In Figs. 6(a) and 7(a), decodable rates of "Method in [18]+ERT," "General Decoder+ERT," and "General Decoder" are on the same line. The decoded images at BER of 10^{-3} and the target bit rate of 1.0 bpp are displayed in Fig. 8. Figure 9 shows the values for images decoded using the RS(15,13) and RS(15,11) codes. From these results, we can conclude as follows:

- The proposed scheme provides high-quality error correction regardless of the use of the error resilience tools (such as termination, segmentation symbols, and resynchronization markers) in the standard JPEG2000 coding.
- The proposed methods with data hiding schemes (a) and (b) are almost the same quality of error correction.

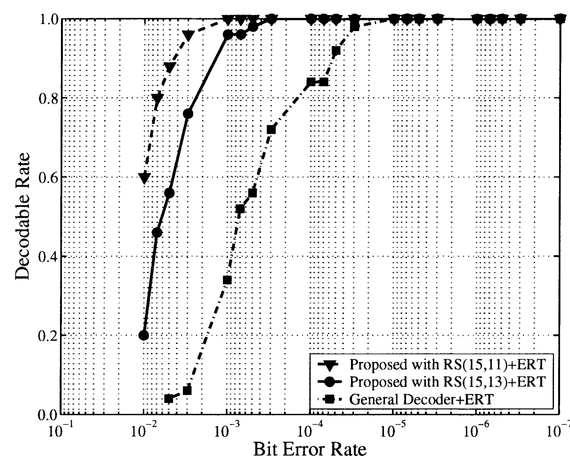
PSNR: 15.86 [dB]
(a) General decoder.

PSNR: 19.28 [dB]
(b) General decoder with ERT.

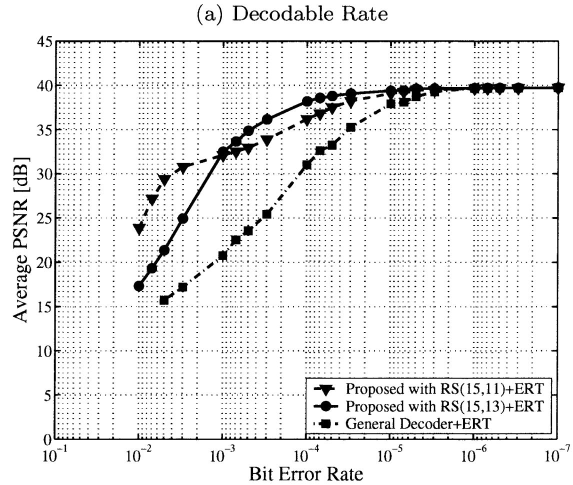
PSNR: 29.58 [dB]
(c) Proposed scheme with data hiding (b).

PSNR: 34.07 [dB]
(d) Proposed scheme with data hiding (b) and ERT.

Fig. 8 Quality of error correction with RS(15,13) at BER of 10^{-3}. (target bit rate: 1.0 bpp)

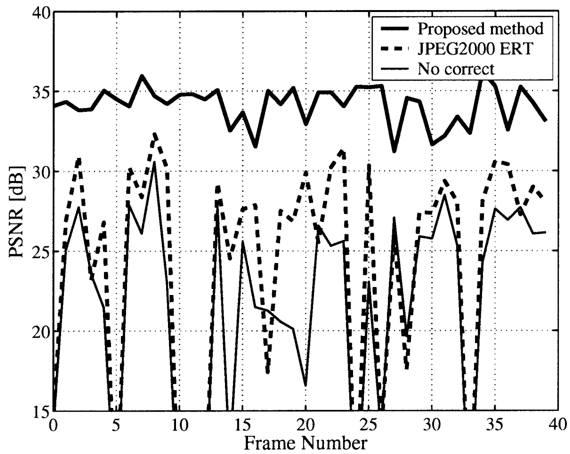


(a) Decodable Rate

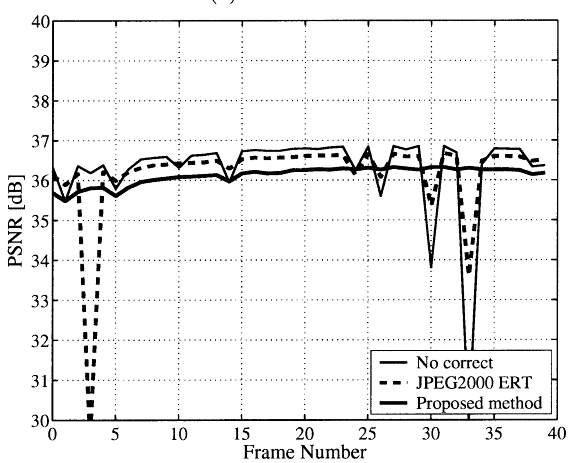


(b) Average PSNR

Fig. 9 Comparison of RS(15,13) and RS(15,11). (target bit rate: 1.0 bpp)



(a) BER : 10^{-4}



(b) BER : 10^{-6}

Fig. 10 Quality of error correction for video sequence (RS(15,13)).

- Under higher BER conditions (more than 10^{-3}), the proposed scheme combined with the error resilience tools is more effective than without the tools and with the tools alone.
- In a high BER environment, RS(15,11) coding provides more protection than RS(15,13) coding. RS(15,13) coding performs better at lower BERs.

(B) Quality of error correction for video sequence

Figures 10(a) and (b) show the error correction capability for the video sequences at the BERs of 10^{-4} and 10^{-6}, respectively.

As shown in Fig. 10, we can see that although the error rate is low, the erroneous headers and higher quality layers cannot be corrected by the standard error resilience tools. Therefore, the image quality is very low. The proposed scheme, however, can correct the errors and highly improve the image quality.

5. Conclusions

We have developed a scheme for correcting errors in JPEG2000 codestream that uses the layer structure. The headers and high quality layers are first FEC-code, then, the parity data is separated from the coded-data and hidden in the LSL of the JPEG2000 codestream. At the decoder, the hidden data is used to correct any errors. Because the generated codestream is compatible with a standard JPEG2000 codestream, it can be decoded by a general JPEG2000 decoder.

Simulation results showed that the data hiding had only a small effect on image quality and that the quality of the error protection was high.

The proposed scheme offers a large freedom to select a location in which the parity data is hidden. For example, a comment area and a marker segment in the main and tile-part headers, and a box in the JPEG2000 file can be used. We plan to consider the selection of the locations and FEC codes in more detail.

Acknowledgement

The authors wish to thank Mr. Takahiro Fukuhara and Mr. Seiji Kimura of Sony Corporation for their useful advice.

This work was supported in part by Grant-in-Aid for JSPS Fellows from the Japan Society for the Promotion of Science.

References

[1] ISO/IEC 15444-1, JPEG2000 Part1 final draft international standard, Aug. 2000.
[2] ISO/IEC 15444-3, Motion JPEG2000 final draft international standard, July 2002.
[3] Information Technology—Generic Coding of Audio-Visual Objects—Part2: Visual, ISO/IEC 14496-2:1999/Amd.1:2000.
[4] D. Taubman and M.W. Marcelin, JPEG2000: Image Compression Fundamentals, Standards, and Practice, Kluwer Academic Publishers, 2001.
[5] A. Skodraqs, C. Christopoulos, and T. Ebrahimi, "The JPEG2000 still image compression standard," IEEE Signal Process. Mag., vol.18, no.5, pp.36–58, Sept. 2001.
[6] I. Moccagata, S. Sodagar, J. Liang, and H. Chen, "Error resilient coding in JPEG-2000 and MPEG-4," IEEE J. Sel. Areas Commun., vol.18, no.6, pp.899–914, June 2000.
[7] T. Fukuhara, K. Katoh, S. Kimura, K. Hosaka, and A. Leung, "Motion-JPEG2000 standardization and target market," Proc. 2000 IEEE Int. Conf. Image Processing (ICIP2000), Vancouver, Canada, Sept. 2000.
[8] A.K. Katsaggelos, F. Ishitiaq, L.P. Kondi, M.-C. Hong, M. Banham, and J. Braiean, "Error resilience and concealment in video coding," Proc. IX European Signal Processing Conf. (EUSIPCO98), pp.221–228, Rhodes, Greece, Sept. 1998.
[9] D.L. Robie and R.M. Mersereau, "Video error correction using steganography," EURASIP J. Applied Signal Processing, no.2, pp.164–173, Feb. 2002.
[10] W.M. Lam, A.R. Reilbman, and B. Liu, "Recovery of lost or erroneously received motion vector," Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP93), vol.5, pp.417–420, Minneapolis, Minnesota, April 1993.
[11] J.W. Park, J.W. Kim, and S.U. Lee, "DCT coefficients recovery-based error concealment technique and its application to the MPEG-2 bit stream error," IEEE Trans. Circuits Syst. Video Technol., vol.7 no.6, pp.845–854, June 1997.

[12] P. Salama, N.B. Shroff, and E.J. Delp, "Error concealment in MPEG video streams over ATM networks," IEEE J. Sel. Areas Commun., vol.18, no.6, pp.1129–1144, Istanbul, Turkey, June 2000.
[13] M. Kurosaki and H. Kiya, "Error concealment using a data hiding technique for MPEG video," IEICE Trans. Fundamentals, vol.E85-A, no.4, pp.790–796, April 2002.
[14] B.A. Banister, B. Belzer, and T.R. Fischer, "Robust image transmission using JPEG2000 and turbo-codes," IEEE Signal Process. Lett., vol.9, no.4, pp.117–119, April 2002.
[15] A. Natu and D. Taubman, "Unequal protection of JPEG2000 codestreams in wireless channels," Proc. IEEE Global Telecommunications Conf. (GLOBECOM2002), vol.1, pp.534–538, Taipei, Taiwan, R.O.C., Nov. 2002.
[16] S.D. Rane, J. Remus, and G. Sapiro, "Wavelet-domain reconstruction of lost blocks in wireless image transmission and packet-switched networks," Proc. Int. Conf. Image Processing, Rochestre, New York, Sept. 2002.
[17] L. Atzori, S. Corona, and D.D. Giusto, "Error recovery in JPEG2000 image transmission," Proc. Int. Conf. Acoustics, Speech. Signal Processing, Salt Lake City, Utah, May 2001.
[18] M. Kurosaki and H. Kiya, "Error concealment using layer structure for JPEG2000 images," IEICE Trans. Fundamentals, vol.E86-A, no.4, pp.772–778, April 2003.
[19] K. Munadi, M. Kurosaki, K. Nishikawa, and H. Kiya, "Error protection for JPEG2000 encoded images and its evaluation over OFDM channel," Proc. IEEE Int. Symp. Circuits and Systems (ISCAS2003), vol.II, pp.432–435, Bangkok, Tailand, May 2003.
[20] D. Taubman, "High performance scalable image compression with EBCOT," IEEE Trans. Image Process., vol.9, no.7, pp.1158–1170, July 2000.
[21] JPEG2000 Verification Model 8.6 Software, ISO/IEC JTC1/SC29/WG1 N1894.
[22] I.S. Reed and G. Solomon, "Polynomial codes over certain finite fields," J. Soc. Industrial and Applied Mathematics, vol.8, pp.300–304, June 1960.
[23] K. Ando, H. Kobayashi, and H. Kiya, "A method for embedding binary data into JPEG2000 images using layer function," IEICE Trans. Inf. & Syst. (Japanese Edition), vol.J85-D-II, no.10, pp.1522–1530, Oct. 2002.

Masayuki Kurosaki was born in Hiroshima, Japan, on November 13, 1977. He received the B.E. and M.E. degrees in electrical engineering from Tokyo Metropolitan University, Tokyo, Japan, in 2000 and 2002, respectively. He is currently a Ph.D. candidate at Tokyo Metropolitan University. His research interest is in image processing. He is a student member of the IEEE.

Hitoshi Kiya was born in Yamagata, Japan, on November 16, 1957. He received the B.E. and M.E. degrees in electrical engineering from Nagaoka University of Technology, Niigata, Japan, and the D.E. degree in electrical engineering from Tokyo Metropolitan University, Tokyo, Japan, in 1980, 1982, and 1987, respectively. In 1982, he joined Tokyo Metropolitan University, where he is currently a Professor of Electrical Engineering, Graduate School of Engineering. He was a visiting researcher of the University of Sydney in Australia from Oct. 1995 to March 1996. His research interests are in digital signal processing, multirate systems, adaptive filtering, image processing, and efficient algorithms for VLSI implementation. He is an Associate Editor of Trans. of IEICE and served an Associate Editor of Trans. on Signal Processing of IEEE from 1998 to 2000. He is a Senior Member of the IEEE, and a member of the Image Electronics Engineers of Japan and the Institute of Television Engineers of Japan.