Error Detection and Concealment in JPEG Images

Mourad ABDAT, Ziad ALKACHOUH and Maurice G. BELLANGER
CNAM, 292 rue Saint-martin, 75141 PARIS CEDEX 03, FRANCE
Tel. : +33 1 40 27 20 82, Fax. : +33 1 40 27 27 79, e-mail: abdat@cnam.fr

ABSTRACT

Even with use of restart intervals, some residual errors remain in the decoded JPEG images after transmission. In order to improve the image quality, robust decoding techniques are useful. First, we propose error detection techniques, then error compensation and concealment techniques for the damaged blocks. Depending on the entropy coding and on the neighbourhood template, improvements between 3 and 7 dB in terms of Peak-to-peak Signal-to-Noise Ratio (PSNR) are provided by robust decoders with respect to conventionnal JPEG decoders, under bit error rates around and less than 10^{-4}.

1 Introduction

The JPEG (Joint Photographic image Experts Group) DCT based coding schemes operate as follows : The original image is scanned sequentially in 8×8 pixel blocks which are DCT (Discrete Cosine Transform) transformed. The DCT coefficients are then quantized using provided quantization tables. The DC quantized coefficient is differentially coded and the AC coefficients are rearranged in a zigzag order and then they are entropy coded with a runlength-Huffman encoding for the baseline process and with a binary arithmetic coder for the extended process [1, 5].

Transmission errors in entropy coded data propagate over several DCT coefficients and blocks. This propagation is mainly due to entropy coding, runlength encoding, differential DC coding and inverse DCT for the spatial domain.In order to limit error propagation, a simple approach is to use independant synchronization points at regular user-fixed interval lengths. At the encoder, the DPCM coding and the entropy encoder are reset at the begining of each interval, and the same is done at the decoder side upon the reception of the synchronization markers. This confines the errors in the interval where they occured. The different intervals are independent, thus, they are also usefull for parallel decoding of the image [2]. In the case of the JPEG standard the synchronization intervals are named restart intervals. Even with the use of restart intervals, some residual errors remain in the decoded images. In order to reduce these errors, we have proposed in a recent work detection and concealment techniques of the residual errors in the case of the baseline process with one neighbourhood template [3]. In this paper, we propose an improvement to the detection techniques and a new neighbohood template. An adaptation of the proposed solutions to the extended coding process with arithmetic coding is also introduced.

The paper is organized as follows : Section 2 presents the residual error detection techniques. The error concealment method is presented in section 3. The objective and subjective results obtained for the detection and concealment techniques in the case of the two entropy coding schemes with different neighbourhood templates are given in Section 4. The conclusion discusses some further work and extensions of the proposed methods.

2 Error detection

2.1 Error detection by coherence tests

The first coherence level which is tested is relative to the transfer format markers and segments. An incoherence in the received transfer format structure is considered as an error. Concerning the coherence tests at the block and entropy coding levels, they depend on the entropy coding technique used. In the case of the baseline process, three coherence levels are considered. The first one deals with the received Huffman codewords. In some cases, it happens that the decoder reads words whose length is exceeding the maximal permitted length for Huffman codewords (16bits). In this case, it is clear that the received word is not recognized in the Huffman table. The second level is at the runlength decoding, it consists of detecting incoherent values of SIZE and incoherent successions of (RUN,SIZE) values like the succession of two EOBs. At the block level, it occurs, under certain errors, that the number of decoded coefficients in a block is greater than 64, this corresponds to an error.

Concerning the arithmetic coder, only coherence tests at the transfer format structure level are considered. The decoder can not detect incoherences in the arithmetic entropy received bitstream.

2.2 Error detection in the frequency domain

This method is intended for error detection in the AC coefficients. It compares the amplitudes of the current block AC coefficients with those of the neighbouring blocks. Taking into acount the AC coefficient distribution [6], the comparison is based on the hypothesis that frequency active zones in the images are not isolated. It is rare to find an isolated AC coefficient with a large amplitude, there will be at least an adjacent block with a similar AC coefficient at the same rank in the zigzag order. As shown in figure 1, two types of block neigborhoods are used.

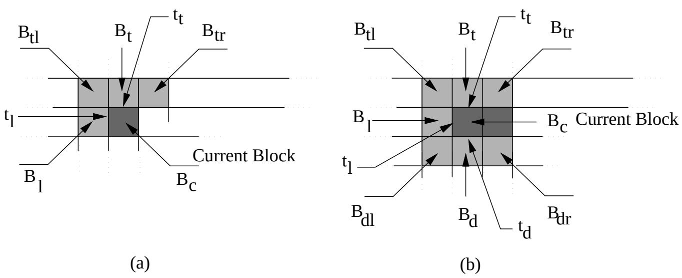

Figure 1: Neighbourhoods used for error detection and concealment. (a) : Prediction neighbourhood. (b) : Interpolation neighbourhood

The prediction neighbourhood uses 4 surrounding blocks. If k is the rank of the current AC coefficient in the current block, the coefficient is declared errouneous if one of the two following conditions holds :

$$|B_c(k)| \geq \alpha\left(max\left(|B_l(k)|, |B_t(k)|, |B_{tl}(k)|, |B_{tr}(k)|\right) + 1\right) \quad (1)$$

or

$$|B_c(k)| \leq \beta\left(min\left(|B_l(k)|, |B_t(k)|, |B_{tl}(k)|, |B_{tr}(k)|\right) + 1\right) \quad (2)$$

The functions max and min return respectively the maximum and minimum of their arguments. The coefficients α and β depend on the rank k : $\alpha = \frac{1}{\beta} = k + 1$.
The interpolation neighbourhood uses 7 surrounding blocks. The block to the right is not used because the errors propagate in that direction.The AC amplitude tests are similar to those of the case of the prediction neighbourhood and we use the values :

$$c\_hor(k) = B_l(k) \quad (3)$$

$$c\_vert(k) = \frac{(B_t(k) + B_d(k))}{2} \quad (4)$$

$$c\_diag1(k) = \frac{(B_{tl}(k) + B_{dr}(k))}{2} \quad (5)$$

$$c\_diag2(k) = \frac{(B_{tr}(k) + B_{dl}(k))}{2} \quad (6)$$

In the equations 1 and 2 we replace $B_l(k)$, $B_t(k)$, $B_{tl}(k)$ and $B_{tr}(k)$ respectively by $c\_hor(k)$, $c\_vert(k)$, $c\_diag1(k)$ and $c\_diag2(k)$.

2.3 Error detection in the spatial domain

This method is mainly for DC coefficient error detection. Errors on DC coefficients correspond to discontinuities in adjacent block boundaries. Taking into acount that pixel value evolution in natural images is smooth, a hard variation is considered as an error.
Let B_1 and B_2 be two blocks with a shared boundary and let m_1 and m_2 be respectively the average values of the pixels of B_1 and B_2 boundaries and σ_1^2, σ_2^2 their variances. The comparison of the frontiers of the two adjacent blocks is based on the value :

$$t = 2\frac{|m_1 - m_2|}{\sigma_d} \; where \; {\sigma_d}^2 = \frac{{\sigma_1}^2 + {\sigma_2}^2}{2} \quad (7)$$

According to [7], the optimal value of t to detect natural edges with 8 pixels in both sides of the edge is : $t_{opt} = 2.92$. Thus, an edge is detected if $t > t_{opt}$. In order to preserve natural edges but detect transmission errors, the threshold must be higher than t_{opt}. We have adopted a dynamic threshold which takes into acount the image context :

$$t_{threshold} = t_m(last\ correct\ block) + t_{opt} \quad (8)$$

where $t_m(last\ correct\ block)$ is the average of the t values of the boundaries of the last correctly decoded block. The initial value of the threshold corresponds to twice t_{opt}.

As shown in figure 1, two neighbourhood templates are used, they correspond respectively to two or three block boundaries. For the prediction neighbourhood, an error is detected if $t_t > t_{threshold}$ and $t_l > t_{threshold}$. In the case of the interpolation neighbourhood, a third boundary is tested : $t_d > t_{threshold}$.
The detection methods presented here are used in the following order : coherence tests, frequency domain detection and then spatial domain detection.

3 Error concealment and error compensation

In the case of the baseline process, when an error occurs, in most cases, the decoder recovers synchronization after decoding few blocks. The remaining blocks in the restart interval are not damaged. In order to preserve undamaged blocks in the restart interval, the detection technique returns the type of the detected error. Mainly, we have distinguished merge errors, split errors, incoherences in the entropy data, spatial errors and frequency domain errors.
In the case of the extended process with arithmetic coding, all the errors are processed in the same manner. The current block and remaining ones in the restart interval are concealed.
The damaged block concealment is similar for the two processes. In the case of the prediction neighbourhood

shown in figure 1, the DC coefficient is concealed by the average value of the surrounding DC values, while the AC coefficients of the rank k in the zigzag order is concealed by :

$$B_c(k) = \frac{(maxabs\,(B_l(k), B_t(k)) + maxabs\,(B_{tl}(k), B_{tr}(k))}{2} \quad (9)$$

The function $maxabs$ returns the coefficient which has the maximum absolute value. This concealment takes into acount the horizontal, vertical and diagonal edges of the surrounding blocks.
In the case of the interpolation neighbourhood shown in figure 1, the DC coefficient is concealed by the average of the seven surrounding blocks and the AC coefficients are concealed by the same equation as in 9 by replacing $B_l(k)$, $B_t(k)$, $B_{tl}(k)$ and $B_{tr}(k)$ respectively by $c\_hor(k)$, $c\_vert(k)$, $c\_diag1(k)$ and $c\_diag2(k)$ given in equations 3 through 6.

4 Results

The results presented concern the image "Gold Hill" which is a 720×576 one component image. The image is coded with 15 MCU (blocks) per restart interval, this correponds to a decrease in compression ratio of 1.94% in the case of the baseline process and to 18.60% in the case of the extended process.
Figure 2 ((a) and (b)) presents the obtained PSNRs versus the bit error rate (BER) for the two coding processes with the prediction neighbourhood. For bit error rates around 10^{-4}, the provided PSNR improvement is about $5dB$ in the case of the baseline process. In the case of the extended process, the levels of PSNRs are less interesting than in the baseline case. The improvement provided by our decoder with respect to the conventional JPEG decoder is about 3 to $4dB$ for all the bit error rates.

Figure 2 ((c) and (d)) gives the results of the interpolation neighbourhood case. The improvement are more significant than those obtained in the case of the prediction neighbourhood. More than $5dB$ are provided by our robust decoder in the case of the baseline process. In the case of the extended process the gain is about $5dB$.

In order to complete the objective results, figure 3 presents visual results obtained for the "Gold Hill" image under bit error rate of 0.210^{-3} and a 15 MCU restart interval length. Clearly, the quality of the images decoded by the robust decoders is better than that of the images decoded by JPEG conventional decoders. The interpolation neighbourhood shows better results than the prediction neighbourhood.

5 Conclusion

Considering the results obtained, it appears that the baselineprocess is best suited for JPEG image transmission over error-prone media. As a recommendation for practical applications, we suggest a restart interval length around 20MCU for the baseline process and a higher length, preferably guided by the compression/robustness trade-off in the case of the extended process with arithmetic coding. For a fast implementation we recommend the error detection and concealment methods with a prediction neighbourhood. For best quality of decoded images, we recommend the interpolation neighbourhood.
Work is now in progress to improve the concealment method and the detection techniques in the case of the extended process. The adaptation of the proposed techniques to animated images, namely for the MPEG standard, is also under consideration.

References

[1] ISO/IEC, IS 10918, UIT-T, rec. T.81, "Information technology - Digital compression and coding of continuous-tone still images -", Part 1 : Requirements and Guidelines, 1992.

[2] M. Abdat and M. Bellanger, "Three Image JPEG Like Coding Schemes and their Robustness to Transmission Errors", *EUSIPCO 1994*, Edinburgh (Scotland), September 1994.

[3] M. Abdat and Z. Al Kachouh, "Masquage des Erreurs de Transmission pour les Images JPEG", *Colloque GRETSI*, Juan-les-pins (FRANCE), September 1995.

[4] M. Abdat, "Etude des Techniques de Compression des Images Fixes et Amélioration de la résistance aux Erreurs de Transmission", *Thèse de Doctorat du Conservatoire National des Arts et métiers*, Paris, Décembre 1995.

[5] W. B. Pennebaker and J. L. Mitchell, "JPEG : Still Image Data Compression Standard", *Van Nostrand Reinhold*, New York, 1993.

[6] R. C. Reininger and J. D. Gibson, "Distribution of Two Dimensional DCT coefficients for Images", *IEEE Transactions on Communications*, Vol. COM-31, No. 6, June 1983, pp. 835-839.

[7] M. Kunt, G. Granlund et M. Kocher, "Traitement Numérique des images", *Presses Polytechniques et Universitaires Romandes*, Traitement de l'Information : Volume 2, Juillet 1993.

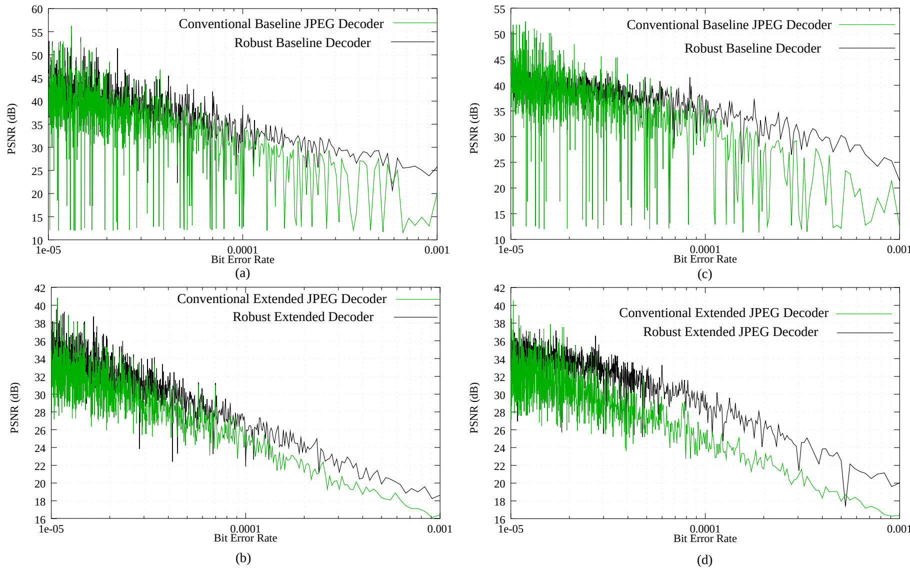

Figure 2: PSNRs obtained by the robust and conventional JPEG decoders for the image "Gold Hill" coded with $15MCU/RST$ (a) : Baseline process and prediction neigborhood (b) : Baseline process and interpolation neigborhood (c) : Extended process and prediction neigborhood. (d) : Extended process and interpolation neigborhood

Figure 3: "Gold Hill" decoded image by the robust and conventional JPEG decoders with a $15MCU/RST$ interval length and under a BER of $0.2\ 10^{-3}$ (o) : Error free decoded image. (a) : Conventional Baseline process, PSNR=25.50dB. (b) : Robust Baseline process with prediction neighbourhood, PSNR=29.94dB. (c) : Robust Baseline process with interpolation neighbourhood, PSNR=31.15dB. (a1) : Conventional extended process, PSNR=21.09dB. (b1) : Robust extended process with prediction neighbourhood, PSNR=23.75dB. (c1) : Robust extended process with interpolation neighbourhood, PSNR=26.29dB