PATTERN-BASED ERROR RECOVERY OF LOW RESOLUTION SUBBANDS IN JPEG2000

Alireza Aminlou, Nasim Hajari, Hossein Badakhshannoory, Mahmoud Reza Hashemi, Omid Fatemi

Multimedia Processing Laboratory, School of Electrical and Computer Engineering,
University of Tehran, Tehran, Iran

ABSTRACT

Digital image transmission is widely used in consumer products, such as digital cameras and cellular phones, where low bit rate coding is required. In any low bit rate encoder, such as the JPEG2000 standard, data truncation (during the encoding process), and data loss (during transmission) will result in lost bit-planes, which will be normally replaced by zeros. In this paper a new algorithm has been proposed, which recovers the lost/truncated lower bit-planes of coefficients in the LL subband of a wavelet transform in a JPEG2000 stream using the data available in higher bit-planes of the same coefficient and its eight neighbors. Simulation results indicate that the proposed algorithm achieves 5.40-8.77 dB improvement with respect to zero filling data recovery method.

***Index Terms*—** Error recovery, JPEG2000, bit-plane coding, data truncation, Discrete Wavelet Transform

1 INTRODUCTION

In recent years, digital image transmission has received considerable attention. The objective is to achieve the highest transmitted quality while minimizing the bit-rate. Part of the original image information is lost during the compression process that is performed for a better usage of the limited bandwidth of the communication channel. More data loss occurs when the compressed image is being subject to channel errors, which is normally the case in wireless transmission.

The JPEG2000 is an emerging image coding standard developed by ISOIIEC Joint Photographic Experts Group (JPEG) [1]. JPEG2000 is able to achieve low bit rates and has provided the required components for error resiliency, making it suitable for wireless image transmission. In JPEG2000, initially, images have to be transformed from RGB color space to the well known YCrCb color components. Each component is wavelet transformed into four different subbands called: LL, HL, LH and HH. In order to perform a multi-level transform, the LL, the low frequency subband, is decomposed into four subbands, recursively. The coefficients of each subband are scalar-quantized and converted into sign and magnitude format. Each subband is further divided into smaller blocks (e.g. 32×32) called codeblock. In the encoding process each codeblock is treated as a set of bit-planes which are encoded using an arithmetic encoder, producing a bit-stream. There is always a trade-off between image quality and bit-stream length. In order to achieve lower bit rates, which naturally lead to a reduced quality, the bit stream is required to be truncated at some points, generating the final bit-stream. The final bit-stream is split into packets of data to be transmitted or stored in a file.

A bit error even in the transmission of a single bit will result in loss of synchronization at the entropy decoder such that the reconstructed image can be fully and severely damaged. In order to prevent drastic reduction in the quality of the decoded image due to simple transmission errors, error resilience tools are developed for JPEG2000. These tools include the use of resynchronization markers, the coding of data in relatively small packets and segmentation symbols. Resynchronization markers enable the decoder to reestablish synchronization when bit errors occur within a packet. The addition of segment symbols within the bit-stream at the end of each bit-plane allows errors within the coded bit-planes of the codeblocks to be detected.

It must be noted that even if the coded image data is transmitted without any errors, the truncation of the encoded data, in order to achieve better compression, by itself introduces a data loss which leads to a reduction in quality. Hence the importance of bit-plane recovery methods at the decoder is further emphasized.

Different methods are used to replace the lost bit-planes of each codeblock. The most well-known method, zero filling, replaces the discarded bit-planes by zeros. Another efficient method, half filling, substitutes a value which is equal to half of the value of the lost data [1].

The existing block-based error recovery methods in [2],[3] and [4] for JPEG/MPEG are not suitable for JPEG2000 [5]. The method proposed in [6] is based on a different assumption that a whole bit-plane of an LL subband codeblock is lost but uses the data available from other subbands for recovery. [7] conceals the lost bit planes using cross subband correlation. The method does not apply to the low frequency subband (LL). Therefore, when the low frequency subband is lost, there is still significant image deterioration. [5] uses *Interesting Direction Sets (IDS)* of in-subbands, which are direction patterns defined separately for each subband, to find whether a lost bit must be zero or one. This method does not apply to LL subband. In [8] and [9], error concealment methods are proposed by using data embedding and data hiding techniques. [10] uses *Unequal Error Protection (UEP)* method for recovering lost data. This method employed *Reed-Solomon (RS)* detection techniques to conceal the lost data by dedicating more redundant bits to the important data.

In this paper, a new method for recovering the lost bit-planes of the LL subband is proposed. The proposed method recovers lost or truncated bits of each coefficient using the available higher bits of that coefficient and its eight neighbors within the same codeblock.

The remaining of this paper is organized as follows. Section 2 describes the proposed method. Section 3 discusses the simulation results followed by conclusions and references.

2 THE PROPOSED ALGORITHM

A compressed image is composed of several codeblocks; each consisting of $n+m$ (e.g. 8) bit coefficients, where n and m are the number of available and lost/truncated bit-planes, respectively. This is shown in Figure 1.

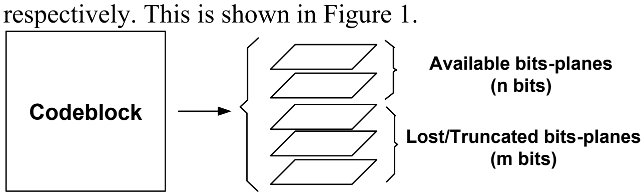

Figure 1. A codeblock, available (n) and lost/truncated (m) bit-planes

In order to recover the lower part (m lower bits) of each coefficient we propose to use the available upper part (n upper bits) of that coefficient and its eight neighbors. We start with the introduction of the main idea behind the new technique, and then proceed to defining the pattern matrix and pattern indexing, the two concepts proposed to help with the implementation of the algorithm.

2-1 The Main Idea: Tendency

A 2-D low frequency discrete signal, like an image or the LL subband of a wavelet transformed image, has a homogeneity property which means that neighboring coefficients values vary from each other by a small amount. Therefore when the upper n bits of a coefficient is less than the upper n bits of its neighbors; the total value of this coefficient tends to be as large as possible in order to be in the nearest vicinity of the value of its neighbors. Hence, since the upper n bits have a fixed value, the lower m bits of that coefficient tend to have its largest value to preserve homogeneity. Similarly, when the neighbors' upper n bits are smaller than the upper n bits of the target coefficient, the lower m bits of that coefficient tend to be as small as possible. Finally, when some of the neighbors' upper n bits have larger values and the others have smaller values compared to the n upper bits of the target coefficient, then the lower m bits of the middle coefficient chooses a value that fits best for preserving the homogeneity. We will refer to this property as "Tendency" which states that the lower m bits of each coefficient tends to a value determined by the upper n bits of that coefficient and its neighbors.

Figure 2 shows the target coefficient $X_{0,0}$ and its eight neighbors. According to Figure 1, the n upper bits of these coefficients are available. Here we define these upper bits of each coefficient X_{ij} as $U_{ij} = X_{ij}/2^m$. The U matrix is then defined as a 3×3 matrix whose elements are U_{ij}. The main idea in this paper is to recover the lower m bits of the target coefficient ($X_{0,0}$) according to different patterns formed by the U matrix. Recovering the lower part of the target coefficient at the decoder is achieved by determining the pattern of the neighboring coefficients and replacing the truncated or lost part by an average value introduced by that pattern. This average is the value that the lower part of the target coefficient tends to according to the tendency property. The latter value is determined experimentally by calculating an average value for each pattern observed in a number of training images.

$X_{-1,-1}$	$X_{-1,0}$	$X_{-1,+1}$
$X_{0,-1}$	$X_{0,0}$	$X_{0,+1}$
$X_{+1,-1}$	$X_{+1,0}$	$X_{+1,+1}$

Figure 2. The target coefficient ($X_{0,0}$) and its neighbors

2-2 The Patterns

Each of the U_{ij} elements can take 2^n different values and therefore the U matrix can have $(2^n)^9$ different states which means $(2^n)^9$ different patterns of data arrangement exist. However, some of them are not distinct ones because of the similarity in their behavior. Similarity between patterns means that some of them with a common characteristic produce approximately the same average value to be replaced by the lower bit-planes of the target coefficient. So they are categorized into one pattern by using a number of observations based on the simulation results.

It has been noted that the value of lower part of the target coefficient is not related to the exact values of upper part of its neighbors; instead it is strongly related to their differences. For instance, if upper parts of neighbors are smaller than upper part of the target coefficient, then the lower part of the target coefficient tends to zero, while the lower part of the target coefficient tends to 2^m -1 when they are greater than the upper part of the target coefficient. It is inferred that the DC value does not affect the average of the patterns. Also, the gradients for negative and positive coefficients in a picture are the same. This is due to the fact that negative coefficients were originally positive values that have been down shifted by a DC value during the encoding process. Therefore, their rate of change is observed to be the same as positive values. Additionally, an image environment is homogenous which means that the gradient of image is directionless. As a result, different patterns with the same amount of changes from the target coefficient but at different directions (horizontal, vertical, diagonal or their combinations) are observed to give a unique behavior and therefore have the same average value.

Based on the above observations, the lower part of the target coefficient ($X_{0,0}$) depends on the relative value of the upper part of its neighbors (U_{ij}). Therefore, we define the pattern matrix D, as shown in Figure 3, for each target coefficient, with values $D_{ij} = (U_{ij} - U_{0,0})$. The D_{ij} elements can take 1, 0 or -1. It should be noted that, if the difference is more than one then it represents an edge like behavior where the lost bits are unpredictable in this case.

$D_{-1,-1}$	$D_{-1,0}$	$D_{-1,+1}$
$D_{0,-1}$	$D_{0,0}$	$D_{0,+1}$
$D_{+1,-1}$	$D_{+1,0}$	$D_{+1,+1}$

Figure 3. Pattern matrix of the target coefficient

During the learning process, the average value for each pattern is calculated by taking the average of the lower parts of the coefficients whose D matrix is equal to that pattern. Each pattern is stored in a database along with its corresponding average value to be used at the recovery process in the decoder. Simulation results show that an average improvement of 7.5 dB with respect to zero filling can be achieved while using this approach.

2-3 Pattern Indexing

Since there are lots of patterns, implementation cost will be high. To make the proposed technique more practical, in this section two indexing methods are proposed that map the average values of different patterns with similar characteristics into a single index. These characteristics can be distinguished using different parameters. A parameter determines the feature characteristic for each pattern. An equation is then used for each parameter to calculate a single average value for patterns with mutual characteristics determined by that parameter. Hence a great number of different patterns are indexed into a single directory and the amount of memory required for preserving the patterns and their average values will be reduced significantly. There is a trade off between complexity of the parameter used and the quality achieved. In this paper, as explained below we propose three parameters, namely: WSum, SMSP and SMSP2.

2-3-1 Weighted Sum

WSum is defined as weighted sum of the upper differences of eight neighbors, as shown in (1), where w_{ij} is the weight of each coefficient and d_{ij} is the upper difference of each neighbor. The effect of each neighbor decreases as its distance from the target coefficient increases therefore, the weight of horizontal, vertical and diagonal neighbors are 3, 3 and 2, respectively. Figure 4 shows the relation between the average value for each pattern (avr_ptrn) and the WSum parameter. To model this relation, the linear regression (2) is used which is suitable for hardware implementation. Simulation results show that by using this indexing method an improvement of 7.31dB with respect to zero filling is achieved.

2-3-2 SMSP and SMSP2

It should be noted that the weighted sum (WSum) does not choose the best characteristic for indexing different patterns. For instance the three distinct patterns of Figure 5 have same weighted sum. It is seen that their actual average values (avr_ptrn) differ a lot. To get more accurate results, we define SMSP parameter as (3). The main idea of this parameter is to distinguish between the individual effects of the +1 and -1 bits in the D pattern matrix. Simulation results show that the effect of the +1 difference value on the pattern average does not increase linearly as the number of +1 value in the D matrix increases. Instead its increase in effect shows a saturating behavior and therefore a parameter based on SQRT model (SMSP) is proposed.

Figure 6 illustrates that the average value of patterns has a better relation with SMSP parameter. Using linear estimation (4) for taking average, an improvement of 7.32 dB is achieved, while applying the square root equation (5), shown in Figure 6, results in an improvement of 7.38 dB with respect to zero filling. The average value defined in (4) is called avr_smsp, while the average in (5) is referred to as avr_smsp2.

$$WSum = \frac{1}{20}\sum w_{ij} d_{ij} \tag{1}$$

$$avr\_wsum = (0.47 + 0.50 \times WSum) \times Q \tag{2}$$

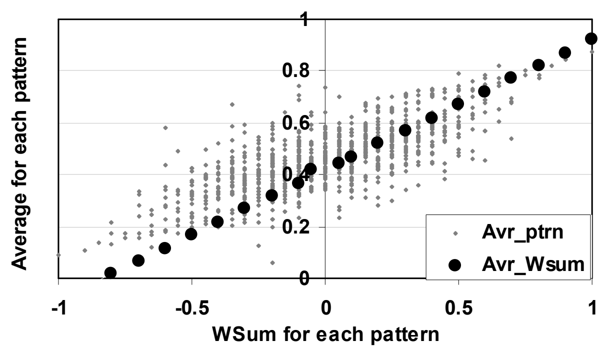

Figure 4. Average of pattern with respect to weighted sum

	Pattern 1			Pattern 2			Pattern 3		
	0	1	0	-1	1	1	-1	1	1
	0	0	0	0	0	0	-1	0	1
	0	0	0	0	0	0	0	0	0
avr_ptrn/2^m	0.663			0.537			0.499		
avr_wsum/2^m	0.556			0.556			0.556		
avr_smsp/2^m	0.546			0.485			0.477		

Figure 5. Three different patterns with same WSum (1/20) and different avr_ptrn

$$smsp = \frac{1}{20}\left(\sqrt{sp} - \sqrt{-sm}\right)^2 \tag{3}$$

$$sp = \sum\sum w_{ij} d_{ij} \quad \Big| d_{ij} = +1$$

$$sm = \sum\sum w_{ij} d_{ij} \quad \Big| d_{ij} = -1$$

$$avr\_smsp = (0.47 + 0.53 \times smsp) \times Q \quad (4)$$

$$avr\_smsp2 = \begin{cases} (0.47 + 0.41\sqrt{+smsp})Q & smsp > 0 \\ (0.47 - 0.41\sqrt{-smsp})Q & smsp < 0 \end{cases} \quad (5)$$

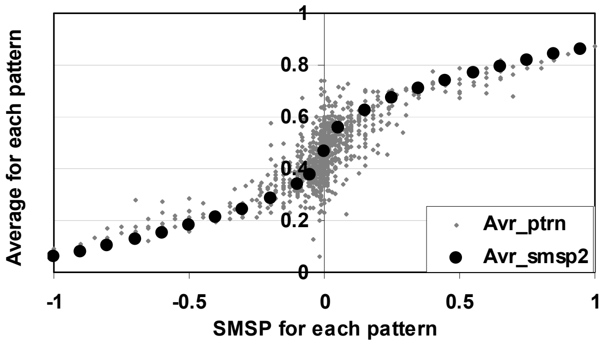

Figure 6. Average of pattern with respect to SMSP

3 SIMULATION RESULTS

To evaluate the proposed new technique we should first estimate these average values of patterns and coefficients of indexing parameters (2), (4) and (5). They were estimated using LL subband of a training set consisting of eight natural images, 256×256 pixels each. The algorithm was then applied to five standard images (Lena, Goldhill, Boat, Peppers, and Parrots) and sixteen natural pictures of 512×512 pixels, using the estimated values. For error recovery, three level wavelet transform is applied to images and lower bit-planes of LL subbands (m=2..7) were truncated. The truncated bit-planes were recovered by zero filling, half-filling and the proposed algorithms. The performance improvement of the algorithms are calculated with respect to that of zero filling and presented in Table 1. The last column shows the average PSNR improvement of the different methods with respect to the zero filling.

PSNR improvements of the proposed methods are 5.40-8.77 dB more than that of zero filling. The method proposed in [5] has an improvement of 1.5-3.0 dB over zero filling, in all quantization factors. When averages of patterns are used for error recovery, an average improvement of 7.5 dB is achieved with respect to zero filling, but it consumes a large amount of memory. The use of avr_smsp2 shows good improvement (7.38 dB in average), in addition to its low hardware resource requirement. However, due to the square root function, its hardware implementation is difficult. Improvement of using avr_wsum is 7.31 dB and it has much simpler hardware implementation due to its linear equation.

Table 1. PSNR improvement with respect to zero filling

m	2	3	4	5	6	7	AVR
Pattern	5.60	7.30	8.45	8.77	8.11	6.75	7.50
smsp2	5.39	7.22	8.40	8.70	8.01	6.57	7.38
smsp	5.40	7.15	8.31	8.60	7.92	6.53	7.32
Wsum	5.40	7.15	8.29	8.59	7.92	6.54	7.31
Half	5.13	6.91	7.76	7.90	7.17	5.83	6.78

4 CONCLUSION

In this paper, a new algorithm has been proposed that recovers truncated or lost bit-planes of a codeblock in low resolution (LL) subbands. Different arrangements of *n* available upper bits of a coefficient and its neighbors have different patterns. Based on the observation from simulation results, these cases are categorized into smaller number of patterns. The proposed algorithm utilizes an average for each pattern to recover the lost/truncated bits instead of using zero or half filling. The experimental results show that the proposed algorithm has the objective results with up to 8.77 improvement over recovery by zero filling. Using the proposed algorithm, the truncated bit planes at the encoder can be recovered and a more error robust image transmission can be achieved using JPEG2000.

REFERENCES

[1] "JPEG2000 part I final draft international standard," *ISO/IEC JTC1/SC29/WG1 N1890, Sept 2000*.

[2] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG 2000 still image coding system: An overview," *IEEE Trans. on Consumer Electronics, Vol.46, No.4, pp. 1103-1127, Nov. 2000*.

[3] I. Moccagatta, S. Soudagar, J. Liang, and H. Chen, “Error-Resilient coding in JPEG 2000 and MPEG-4,” *IEEE Journal on Selected Areas in Communications, Vol.18, No.6, pp. 899-914, June 2000*.

[4] L. Atzori, S. Corona, D.D. Giusto "Error Recovery in JPEG2000 Image Transmission," *IEEE International Conference on Acoustics, Speech, and Signal Processing, Vo1.3, pp.1733-1736, 2001*.

[5] Pei-Jun Lee and Liang-Gee Chen "Error Concealment Algorithm Using Interested Direction for JPEG 2000 Image Transmission" *IEEE Transactions on Consumer Electronics, Vol. 49, No. 4, PP. 1305-1401, Nov. 2003*

[6] Farzad Zargari, Omid Fatemi "Novel Error Concealment Methods in JPEG 2000," *Can. J. Elect Comput. Eng., Vol. 30 No. 2, Spring 2005*.

[7] Pei-Jun Lee, and Liang-Gee Chen "Bit-plane Error Recovery via Cross Subband for Image Transmission in JPEG2000," *Proc. of 2002 IEEE International Conference on Multimedia and Expo (ICME 2002), Vol 1. PP 149-152 August 2002*.

[8] Li-Wei Kang and Jin-Jang Leou, “Two Error Resilient Coding Schemes for Wavelet-Based Image Transmission Based on Data Embedding and Genetic Algorithms*”, IEEE International Conference on Image Processing, Vol 3, pp.:III - 461-464, 2003*.

[9] Jue Wang and Liang Ji, “A Region and Data Hiding Based Error Concealment Scheme for Images” *IEEE Transaction on Consumer Electronics, Vol. 47, No. 2, pp. 257-262, May 2001*.

[10] Ambarish Natu and David Taubman, “Unequal protection of JPEG2000 code-streams in wireless channels”, *Global Telecommunication Conference, GLOBECOM’02, IEEE, Vol. 1, pp. 534-538. Nov. 2002.*