# Bit-Error Aware Lossless Image Compression

Li Tan
Purdue University North Central, Indiana, USA
lizhetan@pnc.edu

Liangmo Wang
Nanjing University of Science & Technology, Nanjing, China
liangmo@mail.njust.edu.cn

## Abstract

In this paper, we propose and investigate simple bit-error aware lossless compression algorithms for the compression and transmission of image data under the bit-error environment. We focus on enhancing two-stage lossless compression algorithms. The first stage uses a simple linear predictor, whereas at the second stage, we apply bi-level block coding, interval entropy coding, and standard entropy coding. The key coding parameters of the predictor, bi-level block coding, or entropy coding parameters are protected by the usage of a forward error correction scheme such as (7,4) Hamming coding. The residues from bi-level bloc coding or the residue offsets from Huffman coding are not protected to compromise the performance of compression ratio. Our compression experiments demonstrate that when the bit error rate (BER) in the channel is equal to or less than 0.001, the lossless compressed image can be recovered with a good quality.

## Introduction

Lossless image compression methods are usually required for compression and transmission of image data whenever a lossy compression approach cannot be applied. Such systems include medical imaging, remote sensing, and high cost archiving systems. Especially in the medical data transmitting and archiving system, the usage of lossy compressed images for diagnostic purposes is prohibited by law in many countries. It is also preferred for the image data of mechanical fault diagnosis to be lossless compressed. In general, a lossless compression algorithm consists of two stages as described in references [1-6]. The first stage performs predictions to remove data correlation in order to produce residue data. The resultant residues have reduced amplitudes and are assumed to be statistically independent with an approximate Laplacian distribution [1-3]. The second stage further compresses residue data using an entropy coding algorithm—that is, Huffman coding or arithmetic coding [1-6]. Much research work has been conducted to improve the compression ratio using a more complex predictor as well as either adaptive Huffman or arithmetic coding with increased algorithm complexity. In addition, the usage of lossless image compression could improve transmission throughput if the compressed image data is transmitted over a network system. However, if bit errors occur in a noisy channel during transmission or in the storage


*ISBN 978-1-60643-379-9*

media, the recovered image will be damaged and will become useless. This outcome results from the fact that a standard entropy coder generates instantaneous codes, which are sensitive to bit errors. Although this problem can be cured by applying a forward error control scheme [7], adding additional bits required by the error correction coding can significantly degrade the performance of the compression ratio and may even cause the expansion of image files. For example, if an 8-bit grayscale image is lossless compressed to 5 bits per pixel, using a (7,4) Hamming code (adding three parity bits for every 4 data bits for a single bit error correction) for bit-error protection will increase the compressed data size by 75%; that is, 8.75 bits per pixel, which indicates image file expansion.

This paper investigates two new simple algorithms: predictive bi-level block coding and predictive interval entropy coding for bit-error aware lossless image compression. To gain a compromised compression ratio, only the prediction parameter, bi-level block coding and interval entropy coding parameters are protected by the (7,4) Hamming codes. The standard predictive entropy coding with bit error correction is also included for comparison purposes. We evaluate and compare the algorithm performances in terms of the compression ratio and the peak signal to noise ratio (PSNR) versus the bit error rate (BER).

## Bit-Error Aware Two-Stage Lossless Image Compression

Figure 1 shows a block diagram of our bit-error aware two-stage lossless compression algorithms. For the first predictive stage, we use up to three neighboring pixels as depicted in Figure 2: the left-hand neighbor (A), the upper neighbor (B), and the upper-left neighbor (C). X is the predicted pixel.

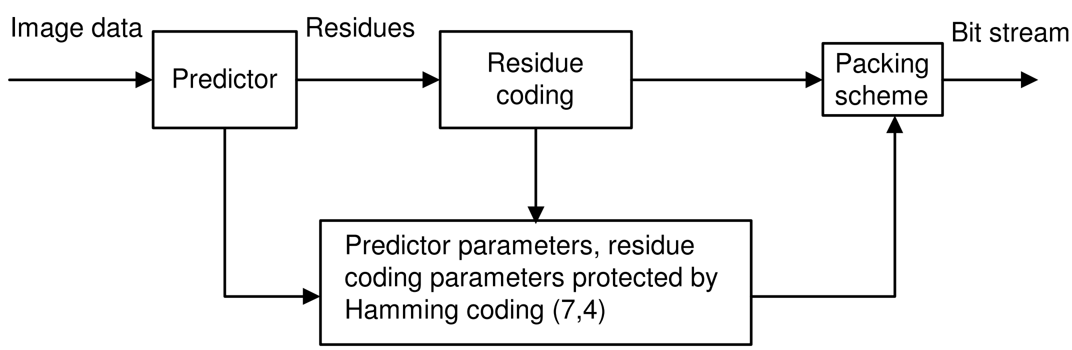



Figure 1: Bit-error aware two-stage lossless image compression scheme


*ISBN 978-1-60643-379-9*

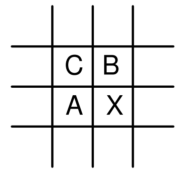



Figure 2: Neighboring pixels for the predictor in an image

Predictive coding is a simple and effective method to remove redundancy of image signals [1-3]. Lossless JPEG [3, 8] contains simple predictors that use the neighboring pixels in Figure 2. The JPEG-LS standard [1] contributes an improvement to lossless JPEG prediction by adding the median filtering process. In addition, the CALIC algorithm [2] offers a slightly better performance of the compression ratio by significantly increasing the amount of computational complexity. For this work, we adopt the linear predictor proposed in [3], which is expressed as

$$P(X) = (3A + 3B - 2C)/4 \qquad (1)$$

This predictor is effective for error resilience, since the predictor output is essentially a weighted sum of neighboring pixels with coefficients less than 1. The predictor parameters required to be stored include the predictor type (3 bits), and the image pixels in the first column and first row each with an 8-bit pixel size. These parameters are further protected using the feed forward error-correction scheme—the (7,4) Hamming coding.

As shown in Figure 1, the second stage is a residue coding stage. We apply bi-level block coding and interval entropy coding methods [9] to encode image residue data line by line. In each line, the key coding parameters are protected using the error control scheme and the residue samples are left as they are to gain a compromise of the compression efficiency. Therefore, the recovered image is no longer lossless when bit errors are introduced by a transmission channel.

**Second Stage Residue Coding**

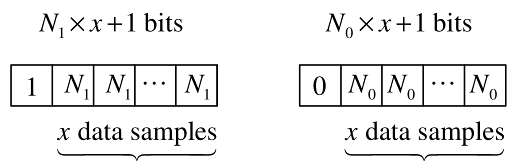
For the residue image obtained from the prediction at the first stage, it is assumed that the redundancy of image signals is removed. The residue samples are assumed to be uncorrelated and to follow the Laplacian distribution approximately. Our objective is to develop residue coding schemes so that they are less sensitive to the bit-error environment.

## A. Bi-level Block Coding for Prediction Errors

We first apply a bi-level block coding scheme [9]. Although it is not as efficient when compared to an entropy coder in terms of lossless compression, it is more robust to bit-errors. Furthermore, after applying the feed forward error control scheme, we can achieve better compression efficiency, since the bi-level block coder requires a smaller number of bits by the bit-error protection algorithm than the amount required by the entropy coder. Assuming that we code the image residue sequence line by line, the coding rules are given in Table 1.

Table 1: Bi-level block coding rules

1. Divide a line of residue data with a length of $n = m \times x$ into $m$ blocks, in which each block consists of $x$ samples; that is, $x$ is the block size. There are two types of blocks: the level-0 block and the level-1 block.




a. Level-1 block     b. Level-0 block

2. For a level-1 block, any sample in the block requires only $N_1$ bits ($N_1 < N_0$ [original sample size]) to encode. Encode each sample using $N_1$ bits and add the prefix "1" to designate the block as the level-1 block.

3. For a level-0 block, at least one of the samples in the block needs more than $N_1$ bits to encode. Encode each sample in the block using $N_0$ bits and add the prefix "0" to indicate the level-0 block.

As shown in Table 1, there are two types of residue blocks, as indicated by a prefix "1" (level-1 block) and prefix "0" (level-0 block), respectively. In the level-1 block, we assume that each sample in the block requires $N_1$ bits to encode, whereas for the level-0 block, we assume that at least one data sample requires $N_0$ bits, where $N_1 < N_0$. We expect that the probability of level-1 blocks is much greater than the probability of level-0 blocks. Hence, the block size $x$ and level-1 sample size of $N_1$ need to be determined optimally to achieve coding efficiency. Considering that the probability of the level-1 block is $P_1 = p^x$, the probability of the level-0 blocks becomes $P_0 = 1 - P_1 = 1 - p^x$, where $p = 1 - p_0$ is the probability of a data sample requiring less or equal to encode using $N_1$ bits, and $p_0$ (which is close to zero) is the probability of a data sample requiring more than $N_1$ bits and less than or equal to $N_0$ bits to encode. For a image residue sequence containing $m$ blocks in which there are $k$ level-1 blocks and ($m - k$) level-0 blocks, the coding length and its probability are, respectively, given below:

$$L(k) = m + N_0 x(m-k) + N_1 xk \qquad (2)$$

$$P(k) = \binom{m}{k} P_1^k (1-P_1)^{m-k} = \binom{m}{k} p^{xk} (1-p^x)^{m-k} \qquad (3)$$

We can obtain the average total length $L_{ave}$ as


*ISBN 978-1-60643-379-9*

$$L_{ave} = \sum_{k=0}^{m} P(k)L(k) = (m + N_0 xm) - (N_0 - N_1)xmp^x \tag{4}$$

By minimizing the average length for a fixed $N_1$, the optimal coding parameters for $x^*$ and $N_1$ can be searched according to the algorithm listed in Table 2. The derivation of the algorithm can be found in [9].

Table 2: Algorithm for searching the optimal coding parameters

1. Find $N_0$ for a given data sequence. Initially, set $N_1 = N_0 - 2$ and $x^* = 4$.
2. For $N_1 = 1, 2, 3, N_0 - 1$

   Estimate $p_0$, the probability of the sample requiring more than $N_1$ bits to encode, and calculate the optimal block size:

   $$x^* = 1/\sqrt{(N_0 - N_1)p_0}$$

   Round up the block size to an integer value.

   If $x^* \times p_0 \le 0.3$, calculate the average bits per sample:

   $$\left(L_{ave}/n\right)_{\min} = 2\sqrt{(N_0 - N_1)p_0} + N_1$$

   Record $N_1$ and $x^*$ values for the next comparison

   End loop

After completing search loops, select $N_1$ and $x^*$ corresponding to the smallest value of $\left(L_{ave}/n\right)_{\min}$.

The data format of predictive bi-level block coding is proposed in Figure 3. As shown in Figure 3, the packing scheme packs the predictor type and coding parameters, which are further protected using the (7,4) Hamming codes as a header. After prediction, the residue coding process operates line by line. Similarly, the bi-level block coding parameters are protected by Hamming coding followed by the unprotected encoded residue bit steam.

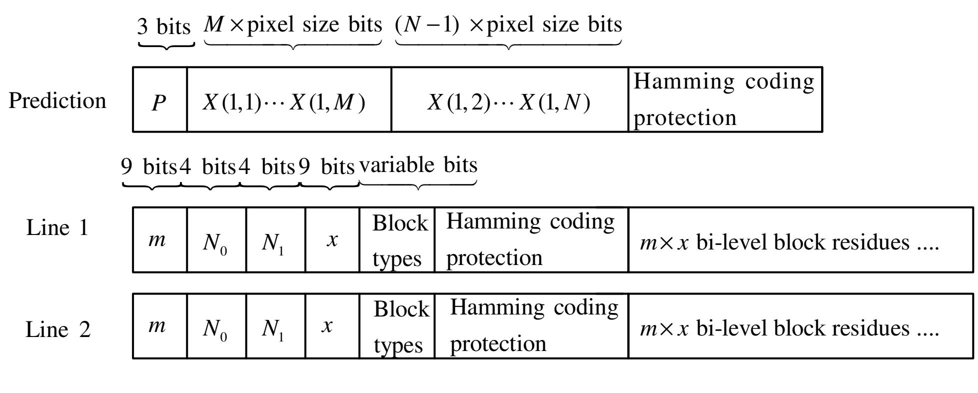



Figure 3: Data format of predictive bi-level block coding

### B. Interval Entropy Coding for Prediction Errors

The interval Huffman coding (its arithmetic version can similarly be developed) can be considered as an alternative method for the second stage image residue coding. For interval entropy coding, we divided each residue into the residue interval and defined its offset portion (residue offset) below:

$$q(n) = floor[r(n)/2^{(N_0-N_1)}] \quad (5)$$

$$offset = r(n) - 2^{(N_0-N_1)} \times q(n) \quad (6)$$

where each interval (symbol) $q(n)$, which is quantized from a residue $r(n)$, is entropy encoded and error protected like the coding parameters, leaving the offset bits as they are. Function $floor(x)$ runs $x$ into the nearest integer towards negative infinity. It has been shown that the entropy coding can only compress approximately one to two bits per sample of a perfect Laplacian sequence [6]. We assume that our entropy coder achieves $N_0 - \beta$ bits per pixel, where $\beta \approx 1-2$ bits. Assuming that $q(n)$ follows a perfect Laplacian distribution, choosing the smaller symbol size $N_1$ for the interval entropy coder will gain approximately the same compression performance. In this work, we choose $N_1 = 3$, since it gives the best results in our experiments. The interval Huffman codes are listed in Table 4. The data format for predictive interval Huffman coding is depicted in Figure 4.

Table 4: Interval Huffman codes

| q(n) | Interval codes | q(n) | Interval codes |
|---|---|---|---|
| 0 | 1 | +2 | 01011 |
| -1 | 00 | -3 | 010100 |
| +1 | 011 | +3 | 010110 |
| -2 | 0100 | -4 | 0101011 |

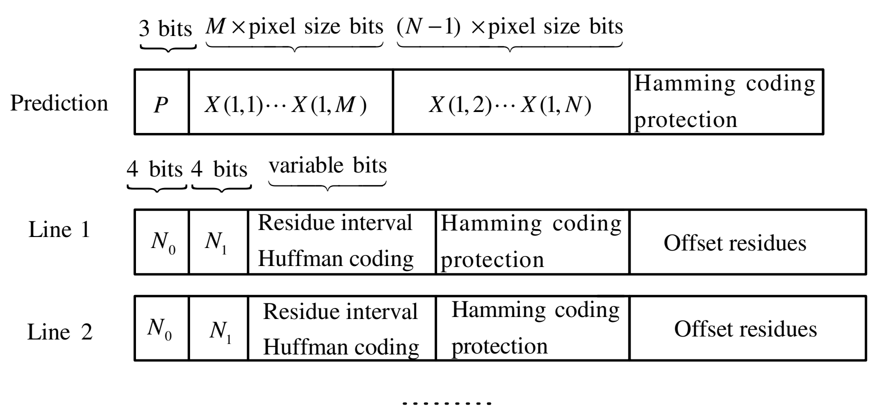



Figure 4: Data format of predictive interval entropy coding

## C. Standard Huffman Coding for Prediction Errors

For comparison purposes, we also include a standard Huffman coding scheme as shown in Table 5 for residue coding. These Huffman codes coincide with the first nine (9) lines in the baseline JPEG algorithm for compressing the DC coefficients [8,10]. Each image residue is encoded using a prefix for code word size as shown in Table 5, cascaded by the binary amplitude bits. For example, to encode -3,-2,+2, and +3, we have **011**00, **0110**1, **011**10, and **111**11, respectively. For our experiment, the only codeword size bits are protected using (7, 4) Hamming codes. The data format is similar to that of Figure 4.

Table 5: Huffman codes

| Code size (No. bits) | Amplitude code | Code size (no. bits) | Amplitude code |
|---|---|---|---|
| 00 (0) | 0 | 110 (5) | -31,…,-16,+16,…,+31 |
| 010 (1) | -1,+1 | 1110 (6) | -63,…,-32,+32,…,+63 |
| 011 (2) | -3,-2,+2,+3 | 11110 (7) | -127,…,-64,+64,…,+127 |
| 100 (3) | -7,...,-4,+4,…,+7 | 111110 (8) | -255,…,-128,+128,…,+255 |
| 101 (4) | -15,...,-8,+8,…,+15 | 111110 (9) | -512,….,256,+256,…,+512 |

## Performance Evaluation and Comparisons

We apply our proposed bit-error aware lossless compression algorithms to eight bit grayscale images ("Lena" and "Boat"), each with the image size of 512x512 pixels. To evaluate the performances for the bit-error environment, in addition to using the average bits per pixel (ABPP) and compression ratio (CR), we also use the peak signal to noise ratio (PSNR in dB) as an error metric to measure the recovered image quality. The PSNR is defined below:

$$\text{PSNR dB} = 20\times\log_{10}\left(\frac{255}{\text{RMSE}}\right) \quad (7)$$

where RMSE is the root mean squared error given by

$$\text{RMSE} = \sqrt{\frac{1}{M\times N}\sum_{i=1}^{N}\sum_{j=1}^{M}[X(i,j)-\hat{X}(i,j)]^2} \quad (8)$$

Note that $X(i,j)$ represents the original pixel, while $\hat{X}(i,j)$ is the recovered pixel. Figure 5 shows our typical experimental results for compressing the "Lena" image using each algorithm under the bit-error rate (BER) $=5\times10^{-3}$. Figure 5 depicts the compression results for the "boat" image. It can be seen that the recovered image from predictive bi-level block coding matches well with pixel levels when compared to their respective original images.

Typical compression results at BER $=10^{-3}$ for the ABPPs, CRs, and PSNRs are listed in Table 6. For compressing the "Lena" image, the predictive bi-level block coding offers the lowest ABPP of 5.73 bits, the highest CR of 28.38%, and the highest PSNR of 34.24 dB. We achieve similar results for compressing the "boat" image. However, the standard predictive Huffman coding essentially has a data expansion with 8.02 bits per pixel instead of compression.

Figures 6 and 7 show experimental results of the PSNRs versus the BER for coding the "Lena" and "boat" images, respectively. We obtain the final PSNR based on the average of ten (10) independent runs at the given BER. When the BER$>10^{-3}$, the predictive bi-level block coding algorithm offers the highest PSNR, hence producing the best signal quality. On the other hand, when the BER $<10^{-3}$, the predictive interval Huffman coding and predictive Huffman coding algorithms tend to gain a better image quality at the same PSNR level. However, the predictive bi-level block coder still offers a good quality of the recovered image with the highest compression ratio. Similar results are obtained for compressing other types of images in our database.


Figure 3: Compression results using predictive bi-level block coding, predictive interval Huffman coding, and predictive Huffman coding at BER $=5\times10^{-3}$


*ISBN 978-1-60643-379-9*

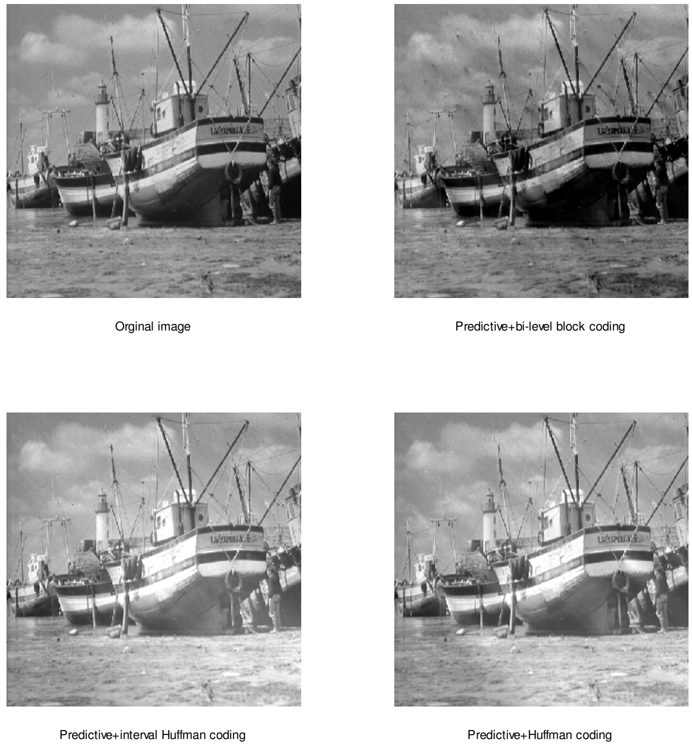
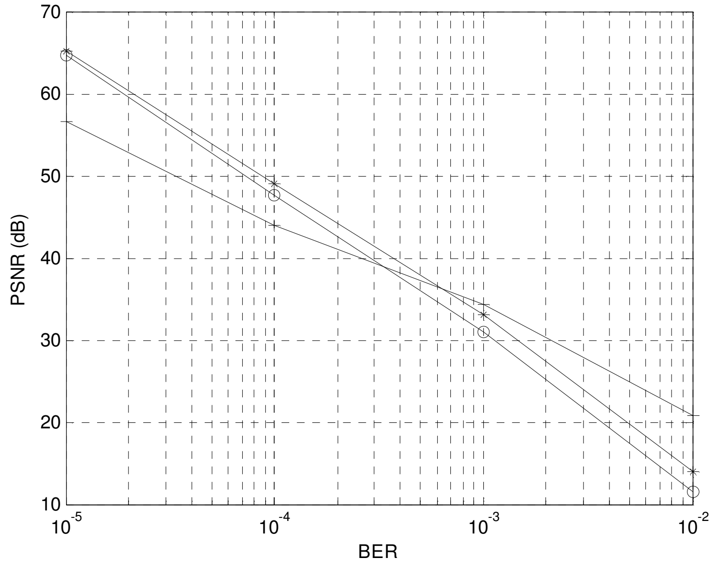



Figure 5: Compression results using predictive bi-level block coding, predictive interval Huffman coding, and predictive Huffman coding at BER $=5\times10^{-3}$

Table 6: Performance Comparisons of Lossless Compression at BER=$10^{-3}$
(Each PSNR is in dB and is obtained by averaging 10 independent runs)

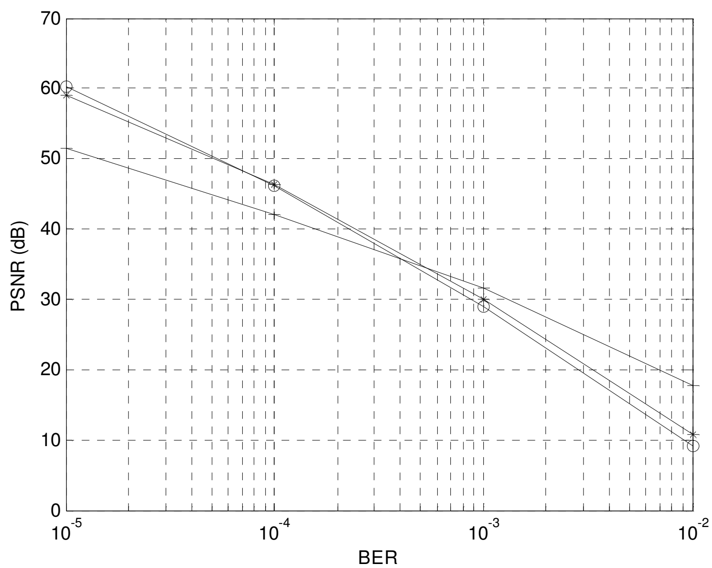
| Algorithms | Image: Lena | Image: Boat |
|---|---|---|
| Predictive bi-level block coding | ABPP: 5.73 bits<br>CR: 28.38%<br>PSNR: 34.24 dB | ABPP: 6.12 bits<br>CR: 23.5%<br>PSNR: 31.72 dB |
| Predictive interval Huffman coding | ABPP: 6.63 bits<br>CR: 17.13%<br>PSNR: 30.89 dB | ABPP: 7.10 bits<br>CR: 11.25%<br>PSNR; 28.98 dB |
| Predictive Huffman coding | ABPP: 7.39 bits<br>CR: 7.63%<br>PSNR: 33.04 dB | ABPP: 8.02 bits<br>CR: Expansion<br>PSNR: 29.97 dB |

.




+: predictive bi-level block coding;
o: predictive interval Huffman coding;
*: standard predictive Huffman coding

Figure 6: PSNR performances versus the bit-error rate for "Lena" image




+: predictive bi-level block coding;
o: predictive interval Huffman coding;
*: standard predictive Huffman coding

Figure 7: PSNR performances versus the bit-error rate for "boat" image


*ISBN 978-1-60643-379-9*

## Conclusion

We have developed the predictive bi-level block coding and interval Huffman coding algorithms for bit-error aware lossless image compression. The prediction parameters at the first stage and the bi-level block coding and interval Huffman parameters are protected by adding (7,4) Hamming error correction codes, thus leaving residues or offset residues as they are to gain a compromised compression ratio. When the bit error rate is larger than 0.001, the developed predictive bi-level block coder offers a better signal quality, as well as the highest compression ratio. On the other hand, when the bit error rate is less than 0.001, the developed predictive interval Huffman coding tends to achieve a higher signal quality. However, the predictive bi-level block coding can still maintain a commendable signal quality with a higher compression ratio.

## References


[1] M. Weinbergner, G. Seroussi and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," IEEE Transactions on Image Processing, Vol. 9, No. 8, August 2000.

[2] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," IEEE Transactions on Communications, Vol. 45, No. 4, April, 1997.

[3] R. Starrosolski, "Simple fast and adaptive lossless compression algorithm," Softw. Pract. xper., Vol. 37, pp. 65-91, 2007.

[4] S. D. Stearns, L. Tan, and N. Magotra, "Lossless compression of waveform data for efficient transmission and storage," IEEE Transactions on Geoscience and Remote Sensing, Vol. 31, No. 3, pp. 645-654, May 1993.

[5] S. D. Stearns, "Arithmetic coding in lossless waveform compression," IEEE Transactions on Signal Processing, Vol. 43, No. 8, pp. 1874-1879, 1995.

[6] S. D. Stearns, L. Tan, and N. Magotra, "A bi-level coding technique for compressing broadband residue sequences," Digital Signal Processing, Vol. 2, No. 3, pp. 146-156, July 1992.

[7] S. Lin, D. Jr. Costello, Error Control Coding: Fundamentals and Applications. Prentice Hall, Inc., Englewood Cliffs, NJ, 1983.

[8] Z. Li, M. S. Drew, Fundamentals of Multimedia. Prentice Hall, Upper Saddle River, NJ 07458, 2004.

[9] L. Tan, J. Jiang, Y. Zhang, "Bit-error aware lossless compression of waveform data," IEEE Signal Processing Letters, Vol. 17, No. 6, pp. 547-550, June 2010.

[10] L. Tan, Digital Signal Processing: Fundamentals and Applications. Elsevier/Academics, 2007.

**Biographies**

LI TAN is currently with the College of Engineering and Technology at Purdue University North Central, Westville, Indiana. Previously, he was a full professor at DeVry University Decatur, Georgia. His research interests are in the areas of active noise control, digital signal processing, digital communications, and mechanical systems and signal processing. He is the author of three textbooks: Digital Signal Processing: Fundamentals and Applications (Elsevier/Academic Press, 2007), Fundamentals of Analog and Digital Signal Processing (AuthorHouse, 2008), and Analog Signal Processing and Filter Design (Linus Publications, 2009), and holds a granted US patent. Dr. Tan is a senior member of IEEE.

LIANGMO WANG is currently a professor at School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing, China. His research interests are in the areas of vehicle dynamics and control, structural optimization, structural dynamics, and signal processing. He has many years of working experience of the automobile industry.