IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 17, 2020, accepted June 4, 2020, date of publication June 9, 2020, date of current version June 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3001073

2-D Bi-Level Block Coding for Color Image Compression and Transmission With Bit-Error Awareness

XUAN PENG[1], **JEAN JIANG**[2], **(Senior Member, IEEE)**, **LIZHE TAN**[1], **(Senior Member, IEEE)**, **AND JINTAO HOU**[1]

[1]Department of Electrical and Computer Engineering, Purdue University Northwest, Hammond, IN 46323, USA
[2]Department of Engineering Technology, Purdue University Northwest, Hammond, IN 46323, USA

Corresponding author: Lizhe Tan (lizhetan@pnw.edu)

ABSTRACT In this paper, we develop a new color image lossless compression algorithm with bit-error awareness based on a general bi-level block coding method. The proposed method contains three stages. First, a color image in the RGB color space is converted to the YCrCb color space by lossless reversible transformations. Next, predictors in the YCrCb color space are applied to generate residue sequences. In order to achieve a better image compression ratio, a particle swarm optimization (PSO) algorithm is adopted to search the best combination from the candidates of color transformations and predictors which generate the minimum residue entropy. Third, a new 2-D bi-level block coding algorithm is developed to further encode the residue sequences. Comparing with the existing residue coding methods, including 1-D bi-level block coding, interval Huffman coding, and standard Huffman coding, the 2-D bi-level block coding algorithm can improve image compression ratio as well as preserving the bit-error resilience. Finally, the key parameters such as color transformation information, predictor parameters and residue coding parameters are protected using (7, 4) Hamming codes in the bit stream before transmission. The performances are validated in terms of compression ratio (CR) and peak signal to noise ratio (PSNR). The compression algorithm with bi-level coding achieves the highest PSNR when the bit-error rate (BER) is larger than 0.001 and maintains an acceptable PSNR for BER less than 0.001. In particular, the compression algorithm using the new 2-D bi-level block coding scheme achieves the highest CR.

INDEX TERMS Lossless compression, bit-error awareness, general bi-level block coding, particle swarm optimization, (7, 4) Hamming code.

I. INTRODUCTION

Digital image information is increasingly adopted in many areas, such as medical imaging, remote sensing applications, biomedical engineering and communication engineering [1], [2]. However, transmission and storage of digital images are facing major challenges due to the growing size of digital image datasets [3]. Due to the enormous data volumes, it is necessary to develop efficient ways to compress images [4]–[7]. Although many lossy compression methods can achieve high compression ratios [8], [9], lossless image compression algorithms are the better choices when it is unsure whether discarding information contained in the image is applicable or not [10], [11]. For example, lossless compression algorithms are often required for ECG signals, 2-D, 3-D or even 4-D medical images [12], [13]. Among the lossless compression approaches, the linear predictive lossless image compression may traditionally involve with prediction which generates the residue sequence and then the residues are compressed by entropy encoding, or run-length coding, or other residue coding algorithms. The linear prediction for 1-D signals using a traditional least-square design method can be found in [14]–[17]. The predictive coding method can remove redundancy between pixels by extracting and encoding the new information in each pixel, where the new information refers to the difference between the current actual value of the pixel and the predicted value of the pixel [18]–[20].

Currently, there are several directions of lossless compression algorithm development, including lossless algorithm

The associate editor coordinating the review of this manuscript and approving it for publication was Zhaoqing Pan.

improvement, data encryption, and lossless deep learning feature compression [21]–[23]. Compression algorithm such as the arithmetic coding method can be improved by using the range adjusting, step size adjusting, and mutual learning scheme [21], where the compression efficiency is mainly focused. In [22], lossless compression algorithm involving with data encryption for high capacity data hiding is proposed. In addition, due to recent advances of hardware technology, the intelligent analysis equipped at the front-end with deep learning becomes practical. Report [23] proposes a strategy to compactly represent and convey the intermediate-layer deep learning features with high generalization capability, to facilitate the collaborating approach between front and cloud ends. Thus, lossless compression of deep learning features demonstrates a promising feasibility, as a series of tasks can simultaneously benefit from the transmitted intermediate layer features. However, the bit stream produced from standard lossless compression algorithms such as Huffman or arithmetic coding is vulnerable to bit errors during transmission, since the bit stream consists of the instantaneous codes [20]. The image quality using such existing lossless compression methods may not be preserved if a noisy transmission environment is involved. A common solution by simply adding a bit-error control scheme [25], [26] to the compressed bit stream may alleviate the problem but often causing a significant reduction of data compression efficiency or sometimes causing data expansion as a result. Such a problem is currently tackled by developing lossless compression algorithms to improve the compression efficiency to compensate data expansion due to using the bit error control scheme; and to possess the feature of bit error resilience [27]–[31], in which the one-dimensional signal and gray-scale image are compressed. However, if the color image compression is considered, there will still exist a room for further improvement.

For color image compression, there are several opportunities for improvement. First, the prediction can be conducted in the YCrCb color space instead of the RGB color space using the reversible color transformation as reported in [24]. The prediction in the YCrCb color space has two advantages [32], that is, the predictors may be more effective in the YCrCb color space; and when an YCrCb image is converted back to an RGB image, there exists a useful feature of bit-error resilience. In addition, choosing the best combination of the image color conversion and prediction can be achieved via an artificial intelligence algorithm such as the particle swarm optimization (PSO) algorithm [33], [34]. Next, an improved residue coding scheme over the existing schemes [27]–[30] can be developed, which may offer a better compression ratio and bit-error resilience. Finally, the error correction for key information can be applied [25], [26] to ensure the recovery process of lossless compression.

This paper is organized as follows to address the above issues. The proposed research work is first described in Section II. In Section III, a generalized bi-level block coding algorithm is derived. Then a new 2-D bi-level block coding scheme is developed as a special case, which offers better data compression while preserving the bit-error resilience. Section IV proposes a three-stage framework for color image compression; and illustrates the determination of the best combination of the color space transformations and the predictors using the particle swarm optimization (PSO) algorithm. Section V outlines the existing residue coding methods including 1-D bi-level block coding, interval Huffman coding, standard Huffman coding, and bit stream using a pack scheme which protects key information parameters with (7,4) Hamming codes [31]. Section VI validates the performances using the new 2-D bi-level block coding algorithm by comparing with the other three residue coding algorithms mentioned above. Finally, the conclusions are presented in Section VII.

This paper has following contributions:

a. Developed a general bi-level block coding scheme with a 2-D bi-level block coding scheme as a special case, which can offer the best compression ratio while preserving bit-error resilience;

b. Developed a three-stage framework for color image compression;

c. Applied the PSO algorithm to determine the best reversible color transformations from the RGB color space to the YCrCb color space and the best predictors in the YCrCb color space.

II. PROPOSED WORK

In the previous work [27]–[30], the compression algorithm consists of prediction and residue coding, where the residue coding is conducted in one-dimension domain assuming the predicted residues are statistically independent. Our motivation and objectives are describes as follows. To improve the previous work, first our proposed work is to generalize 1-D bi-level block coding to N-dimensional bi-level block coding. To compress 2-D residue data, 2-D bi-level block coding, which is a special case of N-dimensional bi-level block coding, can be applied to explore the spatial correlation of 2-D residue data to improve coding efficiency as well as to achieve better bit-error resilience.

As for color image compression, the prediction stage can be split to the lossless RGB to YCrCb conversion and prediction [24]. Our proposed work is to obtain the minimum residue entropy from prediction by a PSO scheme to select the best combination of the conversion formulas and predictors from the RGB color space to the YCrCb color space. While in the recovery process, that is, the conversion from the YCrCb color space back to the RGB color space, we may expect the second-level bit-error resilience. The validation can be easily achieved by comparing the results from applying the related work [27]–[30] by compressing the RGB channel data independently.

III. GENERALIZED BI-LEVEL BLOCK CODING

A. N-DIMENSION BI-LEVEL BLOCK CODING

For an N dimension data samples each with a size of N_0 bits, we can divide data samples into level-1 blocks, in which each sample requires less or equal to N_1 ($N_1 < N_0$) bits to encode, and level-0 blocks, in which at least one sample requires more than N_1 bits and less than or equal to N_0 bits to encode. Assuming all data samples are statistically independent, the probability of a level-1 block can be expressed as $P_1 = p^{x_1x_2\ldots x_N}$, where $x_1, x_2, \ldots, x_N$ are the dimension sizes of a data blocks, respectively. $p = 1 - p_0$ is the probability of a data sample requiring less than N_1 bits to encode and p_0 (close to zero) is the probability of a data sample requiring more than N_1 bits to encode. The probability of a level-0 block can be written as $P_0 = 1 - P_1 = 1 - p^{x_1x_2\ldots x_N}$. For the N dimension data samples consisting of 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:

$$L(k) = m + N_0x_1x_2\ldots x_N(m-k) + N_1x_1x_2\ldots x_Nk \quad (1)$$

$$\begin{aligned} P(k) &= \binom{m}{k} P_1^k\,(1-P_1)^{m-k} \\ &= \binom{m}{k} p^{x_1x_2\ldots x_Nk}\left(1-p^{x_1x_2\ldots x_N}\right)^{m-k} \end{aligned} \quad (2)$$

The average length L_{ave} can be found below:

$$\begin{aligned} L_{ave} &= \sum_{k=0}^{m} P(k)L(k) \\ &= (m + N_0x_1x_2\ldots x_Nm)\sum_{k=0}^{m} P(k) \\ &\quad - (N_0 - N_1)x_1x_2\ldots x_N \sum_{k=0}^{m} kP(k) \\ &= (m + N_0x_1x_2\ldots x_Nm) \\ &\quad - (N_0 - N_1)x_1x_2\ldots x_Nmp^{x_1x_2\ldots x_N} \end{aligned} \quad (3)$$

Assuming that $x_1x_2\ldots x_Np_0 \le \gamma$, where $0 < \gamma \ll 1$, we can approximate probability P_1 using a Taylor series expansion and further omitting the higher-order terms as

$$\begin{aligned} P_1 &= p^{x_1x_2\ldots x_N} = (1-p_0)^{x_1x_2\ldots x_N} \\ &= 1 - p_0x_1x_2\ldots x_N + \ldots \approx 1 - p_0x_1x_2\ldots x_N \end{aligned} \quad (4)$$

Defining $n = x_1x_2\ldots x_N \times m$, we obtain

$$L_{ave} = \frac{n}{x_1x_2\ldots x_N} + nN_1 + (N_0 - N_1)nx_1x_2\ldots x_Np_0 \quad (5)$$

For a fixed N_1, taking derivative of (5) to $x_1, x_2, \ldots, x_N$, respectively and setting the resultant equations to zero lead to the following:

$$\begin{aligned} \frac{\partial L_{ave}}{\partial x_1} &= -\frac{n}{x_1^2x_2\ldots x_N} + (N_0 - N_1)nx_2\ldots x_Np_0 = 0 \\ \frac{\partial L_{ave}}{\partial x_2} &= -\frac{n}{x_1x_2^2\ldots x_N} + (N_0 - N_1)nx_1x_3\ldots x_Np_0 = 0 \\ &\cdots\cdot \\ \frac{\partial L_{ave}}{\partial x_N} &= -\frac{n}{x_1x_2\ldots x_N^2} + (N_0 - N_1)nx_1x_2\ldots x_{N-1}p_0 = 0 \end{aligned}$$

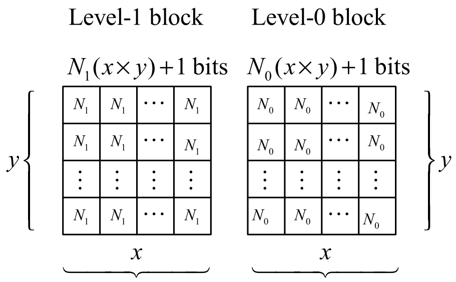
Therefore, we yield the optimal product of block dimensions and the minimum average bits per sample, that is,

$$(x_1x_2\ldots x_N)^* = 1/\sqrt{(N_0 - N_1)p_0} \quad (6)$$

$$\left(\frac{L_{ave}}{n}\right)_{\min} = 2\sqrt{(N_0 - N_1)p_0} + N_1 \quad (7)$$

TABLE 1. 2-D bi-level block coding rules.

1. Divide the data sequence with a length of $n = m\times x\times y$ into m blocks in which each block consists of x columns and y rows, that is, $x\times y$ is the block size. There are two types of blocks, the level-0 block and the level-1 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.

B. TWO-DIMENSION BI-LEVEL BLOCK CODING

By setting $N = 2, x = x_1, y = x_2$ in the N-dimension bi-level block coding, we can develop a 2-D bi-level block coding scheme as shown in Table 1. Equations (1)-(2) are simplified to

$$L(k) = m + N_0xy(m-k) + N_1xyk \quad (8)$$

$$P(k) = \binom{m}{k} P_1^k(1-P_1)^{m-k} = \binom{m}{k} p^{xyk}(1-p^{xy})^{m-k} \quad (9)$$

Correspondingly, Equations (6) and (7) become

$$(xy)^* = 1/\sqrt{(N_0 - N_1)p_0} \quad (10)$$

$$L_{ave} = \frac{n}{xy} + nN_1 + (N_0 - N_1)nxyp_0 \quad (11)$$

$$\left(\frac{L_{ave}}{n}\right)_{\min} = 2\sqrt{(N_0 - N_1)p_0} + N_1 \quad (12)$$

with $xyp_0 < \gamma$. The 2-D bi-level block coding rules are given in Table 1.

The optimal coding parameters are N_1 and $(xy)^*$ corresponding to the smallest $(L_{ave}/n)_{\min}$ through the entire search for $1 \le N_1 \le N_0$. Initially, let $N_1 = N_0 - 2$, and $xy = 4$. Table 2 lists the 2-D bi-level block coding algorithm.

IV. CORLOR TRANSFORMATION AND BEST PREDICTORS

A. THREE-STAGE FRAMEWORK

Fig. 1 depicts our framework for three-stage lossless color image compression.

As shown in Fig. 1, the process of losslessly compressing a color image into a bit stream is divided into three stages.

TABLE 2. 2-D bi-level block coding algorithm.

```
1. Find $N_0$ for a given data sequence.
Initially, set $N_1 = N_0 - 2$ and $(x \times y)^* = 4$.
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:
        $(x \times y)^* = 1/\sqrt{(N_0 - N_1)p_0}$
    Round up the block size to an integer value.
        If $(x \times y)^* \times p_0 \le 0.3$, calculate the average bits per sample:
            $(L_{ave}/n)_{\min} = 2\sqrt{(N_0 - N_1)p_0} + N_1$
        Record $N_1$ and $(xy)^*$ values for the next comparison
    End loop
After completing search loops, select $N_1$ and $(xy)^*$ corresponding to the smallest value of $(L_{ave}/n)_{\min}$.
3. Perform 2-D bi-level block coding using optimal parameters $N_1$, $x$ and $y$ which satisfy $(xy) = (xy)^*$, and rules listed in Table I.
```

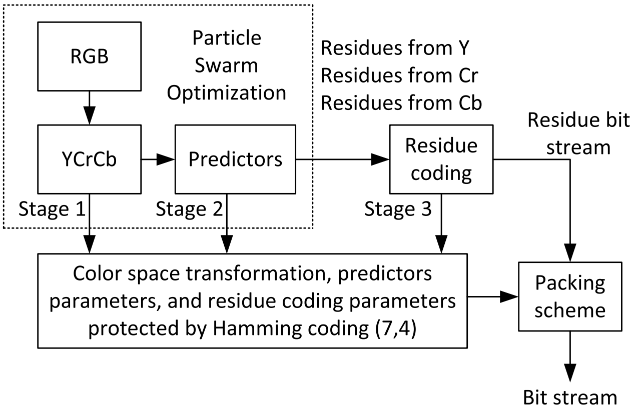


FIGURE 1. Bit-error aware three-stage lossless color image compression.

At the first stage, a RGB image is converted into a YCrCb image, where Y is the luminance component and Cr and Cb are red-difference and blue-difference chrominance components. At the second stage, the YCrCb image is predicted by predictive coders so that the correlation between pixels is reduced and the residue sequence is generated. Note that a particle swarm optimization (PSO) algorithm is used to search the best formulas of color space transformation and the best predictors in the YCrCb color space. The acquired residues are further encoded into a residue bit stream using a residue encoding algorithm at the third stage. In addition, due to bit errors in transmission, some key parameters, that is, the color transformation information, predictor parameters and residue coding parameters, need to be protected by (7,4) Hamming code. The final bit stream is produced by packing these key parameters and unprotected residue bit stream. Decompression is then in a reversed order.

B. COLOR SPACE TRANSFORMATION

YCrCb is used to optimize the transmission of color video and digital signals. The principle is that in the YCrCb format, the luminance channel carries more signal energy, while the chrominance channels carry much less signal energy of color components. After transformation, more effort can be spent on coding the luminance channel [17], which is more predictable. According to this principle, the compression in the YCrCb color space can obtain a higher compression ratio and better bit-error resilience than in the RGB color space. Table 3 lists reversible transformations from the RGB to the YCrCb and vice versa [24], [32], where nine (9) equations are related to conversion from RGB to Y, and twelve (12) equations are related to conversion from RGB to CrCb. As an example for compressing the 512 × 512 "Lena" image on the basis of the maximum compression criterion, the PSO algorithm determines the eighth formula for Y and the first formula for CrCb, that is,

TABLE 3. Transformation formula for Y, Cr and Cb in color space.

i	Y
1	G
2	R
3	B
4	$\lfloor (G+R)/2 \rfloor$
5	$\lfloor (G+B)/2 \rfloor$
6	$\lfloor (R+B)/2 \rfloor$
7	$\lfloor (R+2G+B)/4 \rfloor$
8	$\lfloor (2R+G+B)/4 \rfloor$
9	$\lfloor (R+G+2B)/4 \rfloor$

j	Cr	Cb
1	$R-G$	$B-G$
2	$G-R$	$B-R$
3	$R-B$	$G-B$
4	$R-G$	$B-\lfloor (R+3G)/4 \rfloor$
5	$G-R$	$B-\lfloor (G+3R)/4 \rfloor$
6	$R-B$	$G-\lfloor (R+3B)/4 \rfloor$
7	$B-G$	$R-\lfloor (B+3G)/4 \rfloor$
8	$G-B$	$R-\lfloor (G+3B)/4 \rfloor$
9	$B-R$	$G-\lfloor (B+3R)/4 \rfloor$
10	$R-G$	$B-\lfloor (R+G)/2 \rfloor$
11	$R-B$	$G-\lfloor (R+B)/2 \rfloor$
12	$B-G$	$R-\lfloor (B+G)/2 \rfloor$

$$Y = \lfloor (2R + G + B)/4 \rfloor\,;\ Cr = R - G;\ Cb = B - G. \quad (13)$$

The YCrCb image can be obtained by applying (13) to the RGB image. On the other hand, the converted YCrCb image can also be recovered back to the RGB image as:

$$R = Y + floor\{(2Cr - Cb)/4\} \quad (14)$$

$$G = R - Cr;\ B = Cb + G. \quad (15)$$

For the 512 × 512 "Baboon" image, the transformation formulas for Y at the sixth row, and CrCb at the eleventh row in Table 3 are adopted to obtain a YCrCb image. The YCrCb image can be converted back to the RGB image as follows:

$$G = Cb + Y;\ B = Y - floor\{Cr/2\};$$
$$R = Y + Cr - floor\{Cr/2\}. \quad (16)$$

C. PREDICTION

According to the correlation between pixels, a pixel can be predicted by using the surrounding pixels, and then the difference between the actual pixel value and the predicted value (prediction error) is encoded. If the prediction is correctly chosen, the error will be small, and it can be encoded with fewer bits to achieve data compression. To illustrate, as shown in Fig. 2, X is the predicted pixel; and A, B and C are the

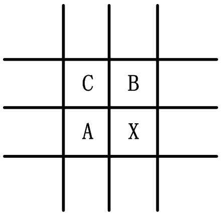


FIGURE 2. Neighbored pixels for the predictor in an image.

TABLE 4. Linear predictors for Y, Cr and Cb.

i	$P(x)$ for Y	$P(x)$ for Cr	$P(x)$ for Cb
1	$(A+B)/2$	$(A+B)/2$	$(A+B)/2$
2	$(3A+3B-2C)/4$	$(3A+3B-2C)/4$	$(3A+3B-2C)/4$

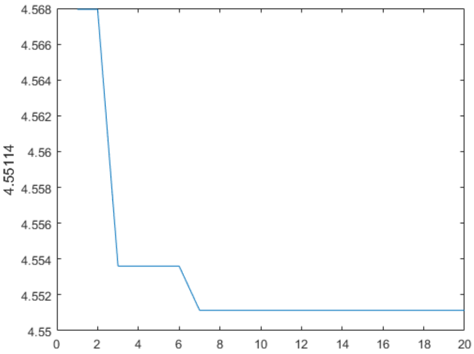


FIGURE 3. Iterative curve of PSO for "Lena" image.

known neighbor pixels of X used for the predictor. According to A, B and C, the value of X can be calculated.

As can be seen in Table 4, two linear predictive equations are selected for Y, Cr and Cb components, respectively [27], [32]. In the light of the maximum compression criterion, the best equations will be adopted.

D. PARTICLE SWARM OPTIMIZATION (PSO)

From Table 3 and Table 4, there are nine formulas of Y space transformation, twelve formulas of CrCb space transformation, and two predictors for Y, Cr and Cb components, respectively. There are 864 combinations. It is tedious to find out the best solution to minimize the compression ratio by checking each combination. The particle swarm optimization (PSO) algorithm can effectively be used to solve this problem. This algorithm uses an evolutionary computation which seeks an optimal solution through collaboration and information sharing among particles in a swarm. The PSO algorithm is listed in Table 5.

Fig. 3 shows the plot of fitness value versus the number of iterations using the PSO algorithm for the 512 × 512 "Lena" image. It takes about 6 minutes to process the image on Lenovo Y900 (i7-6700k). The best fitness value is 4.55114.

TABLE 5. Process of particle swarm optimization.

1. The population size is set to 50, and the number of generations is set to 20.
2. The position and velocity of the particles contain five variables, respectively, where these variables are random. For the position of the particles, the first two variables are the transformation information for converting the RGB image to the YCrCb image. They have 9 and 12 cases respectively, so 4 bits are used for encoding each variable. Each of the last three variables has 2 cases for prediction of Y, Cr or Cb, which is encoded by 1 bit.

4 bits	4 bits	1 bit	1 bit	1 bit
Y	CrCb	Pred Y	Pred Cr	Pred Cb

3. Fitness which consists of five variables of the position is calculated. Fitness value is defined as the information entropy. The formula for calculating information entropy is expressed as

$$\begin{aligned} entropy = &-\sum p_Y(i)\log_2 p_Y(i) \\ &-\sum p_{Cr}(i)\log_2 p_{Cr}(i) \\ &-\sum p_{Cb}(i)\log_2 p_{Cb}(i) \end{aligned}$$

where $p_Y(i)$ is the residue probability from the Y component predictor. $p_{Cr}(i)$ is the residue probability from the Cr component predictor. $p_{Cb}(i)$ is the residue probability from the Cb component predictor.

4. For each particle, its fitness value is compared with the fitness value of the best position it has experienced and the global optimal fitness value. A new individual optimal position and global optimal solution can be obtained.
5. The position and velocity of the particles are changed, where the formulas are given as follows:

$$v(i) = \omega \times v(i) + c_1 \times r_1 \times (pbest(i) - x(i)) + c_2 \times r_2 \times (gbest(i) - x(i))$$
$$x(i) = x(i) + v(i)$$

where ω is the inertia factor. c_1 and c_2, which are set to 2, are learning factors. r_1 and r_2 are random variables from 0 to 1. $pbest(i)$ is the fitness value of the best position every individual has experienced. $gbest(i)$ is the global optimal fitness value.

6. The optimal fitness value obtains through the loop until the maximum number of iterations.

V. RESIDUE CODING ALGORITHM

After prediction, the predicted residues can be compressed sequentially. It is assumed that the residue samples are uncorrelated and follow the Laplacian distribution [14] approximately. Besides 2-D bi-level block coding developed in Section III, 1-D bi-level block coding [30], interval Huffman coding and standard Huffman coding are included for comparison.

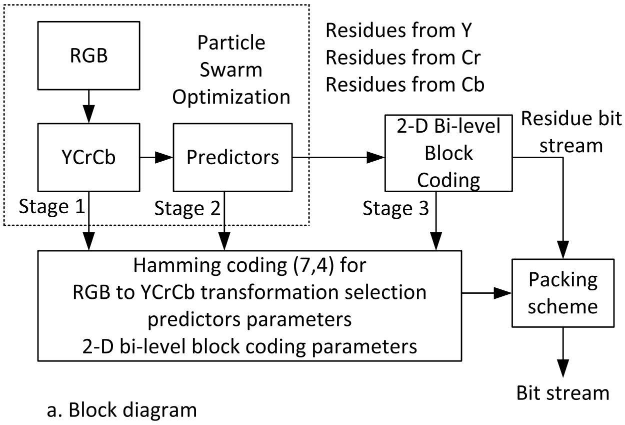
A. BI-LEVEL BLOCK CODING

Based on the 2-D bi-level block coding scheme developed in Section III, our proposed algorithm in a complete form is depicted in Fig. 4.

Note that the 1-D bi-level block coding scheme [27]–[30], is also used for comparison purpose. By using 1-D bi-level block coding, the residue sequence is divided into blocks with a size of x samples and each block is encoded based on the rules in [32].

B. INTERVAL HUFFMAN CODING

Interval Huffman coding is an entropy coding, which can divide the residue into different interval and its offset.



a. Block diagram

Proposed algorithm flow:
(1) PSO minimization of residue entropy (Table V) by
{ selecting RGB to YCrCb transformation,
selecting predictors}
(2) 2-D Block Residue Coding
(3) Hamming coding
{ RGB-YCrCb selection,
predictor selection,
2-D blocking coding key information (Table I)}
(4) Pack
{ Hamming coded bit stream,
residue bit stream}

b. Algorithmic description

FIGURE 4. Bit-error aware YCrCb predictive coding using the new 2-D bi-level block coding scheme.

TABLE 6. Interval Huffman coding.

$q(n)$	Interval codes	$q(n)$	Interval codes
0	1	+2	01011
-1	00	-3	010100
+1	011	+3	0101010
-2	0100	-4	0101011

The interval and offset [30] are determined using the following equations

$$q(n) = floor\{r(n)/2^{(N_0-N_1)}\} \tag{17}$$

$$offset = r(n) - 2^{(N_0-N_1)} \times q(n). \tag{18}$$

where $q(n)$ is the symbol of interval, which is quantized from a residue $r(n)$. It is entropy encoded and error protected. N_0 and N_1 are the symbol sizes. Function $floor(x)$ rounds x down to the nearest integer towards negative infinity. Assuming that $q(n)$ follows a perfect Laplacian distribution, choosing the smaller symbol size N_1 for the interval entropy coder will gain approximately compression. Here choosing $N_1 = 3$ leads the best result from our experiments. The interval Huffman codes are listed in Table 6 [30].

C. STANDARD HUFFMAN CODING

To raise a comparison, standard Huffman coding is also included in this work. The scheme is shown in Table 7 [30].

TABLE 7. Standard Huffman coding.

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)	-511,⋯,-256,+256,⋯,+511

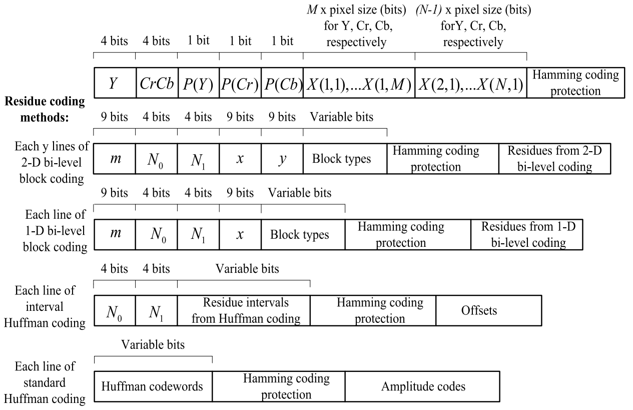

4 bits
4 bits
1 bit
1 bit
1 bit
M x pixel size (bits) for Y, Cr, Cb, respectively
(N-1) x pixel size (bits) forY, Cr, Cb, respectively
Y
CrCb
P(Y)
P(Cr)
P(Cb)
X(1,1),...X(1,M)
X(2,1),...X(N,1)
Hamming coding protection
Residue coding methods:
9 bits
4 bits
4 bits
9 bits
9 bits
Variable bits
Each y lines of 2-D bi-level block coding
m
N0
N1
x
y
Block types
Hamming coding protection
Residues from 2-D bi-level coding
9 bits
4 bits
4 bits
9 bits
Variable bits
Each line of 1-D bi-level block coding
m
N0
N1
x
Block types
Hamming coding protection
Residues from 1-D bi-level coding
4 bits
4 bits
Variable bits
Each line of interval Huffman coding
N0
N1
Residue intervals from Huffman coding
Hamming coding protection
Offsets
Variable bits
Each line of standard Huffman coding
Huffman codewords
Hamming coding protection
Amplitude codes

FIGURE 5. Coding protection for different methods.

Residue of prediction is encoded using a prefix which describes code size, cascaded by the binary amplitude bits. To encode -3, -2, $+2$ and $+3$, for example, the results are 01100, 01101, 01110, and 11111, respectively. The first three numbers are the prefix code. In this work, the prefix part is protected using (7,4) Hamming codes.

For these four residue coding algorithms, key parameters are protected by (7,4) Hamming code. Fig. 5 depicts coding protection for four different residue coding methods.

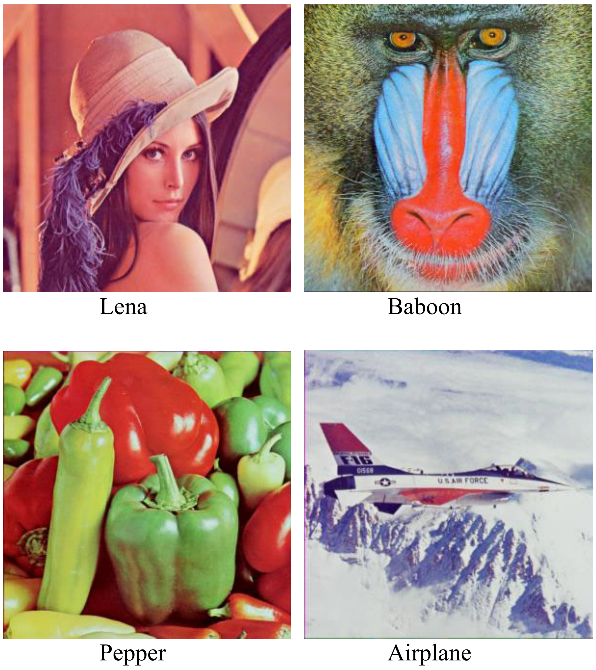
As shown in Fig. 5, (7,4) Hamming code protects color space transformation of Y and CrCb, predictors parameters, first row and first column of the original image, bi-level block coding parameters and block types; interval Huffman coding parameters and residue interval codes, and code words of standard Huffman coding.

VI. RESULT AND DISCUSSION

In our experiments, we adopt four different color images of "Lena", "Baboon", "Pepper", and "Airplane" in the public domain each with a size of 512 × 512, as shown in Fig. 6.

The peak signal to noise ratio (PSNR) and compression ratio (CR) are used to evaluate the image compression and bit-error resilient performance, in which PSNR is related to the image quality. The larger the PSNR, the better the image quality is. The PSNR of RGB image with an $M \times N$ size is given by:

$$PSNR(dB) = 20 \times \log_{10}\left(\frac{255}{RMSE}\right) \tag{19}$$


Lena Baboon

Pepper Airplane

FIGURE 6. Public domain test images.

Note that

$$RSME^2 = \frac{1}{3M \times N}\left(\sum_{i=1}^{N}\sum_{j=1}^{M}[R(i,j)-\hat{R}(i,j)]^2 + \sum_{i=1}^{N}\sum_{j=1}^{M}[G(i,j)-\hat{G}(i,j)]^2 + \sum_{i=1}^{N}\sum_{j=1}^{M}[B(i,j)-\hat{B}(i,j)]^2\right) \tag{20}$$

where $R(i,j)$ and $\hat{R}(i,j)$, $G(i,j)$ and $\hat{G}(i,j)$, and $B(i,j)$ and $\hat{B}(i,j)$ are the original pixels and the recovered pixels, for RGB components, respectively. The CR is the ratio of the pixel size of the original image and the bits of per pixel of the compressed image.

For a RGB color image, it is necessary to convert the RGB image to the YCrCb image. If the image is directly compressed without color space conversion, its compression ratio and image quality will decrease. Table 8 lists the PSNRs and CRs of RGB images without color space conversion as well as the results from the color-converted images using different residue coding algorithms when the bit-error rates (BERs) are 0.001 and 0.005, respectively. Here, we only show the results from the 512 × 512 "Lena" image in details. Each PSNR and CR are obtained by averaging the values from 10 independent runs, respectively. Notice that the PSNR value varies according to the bit error rate while the compression ratio is a robust value.

TABLE 8. Performance evaluation for RGB image and color-converted image.

Algorithms	RGB image	Color-converted image
2-D bi-level block coding	PSNR: 34.3162 dB (BER=0.001) PSNR: 25.8931 dB (BER=0.005) CR: 1.3760	PSNR: 37.5844 dB (BER=0.001) PSNR: 28.9834 dB (BER=0.005) CR: 1.4456
1-D bi-level block coding	PSNR: 35.6342 dB (BER=0.001) PSNR: 26.5518 dB (BER=0.005) CR: 1.3490	PSNR: 37.9850 dB (BER=0.001) PSNR: 28.0733 dB (BER=0.005) CR: 1.4276
Interval Huffman coding	PSNR: 35.4030 dB (BER=0.001) PSNR: 22.1875 dB (BER=0.005) CR: 1.1639	PSNR: 37.1824 dB (BER=0.001) PSNR: 24.0405 dB (BER=0.005) CR: 1.2345
Standard Huffman coding	PSNR: 39.7797 dB (BER=0.001) PSNR: 20.0381 dB (BER=0.005) CR: 1.0532	PSNR: 36.1313 dB (BER=0.001) PSNR: 22.5706 dB (BER=0.005) CR: 1.0735

As shown in Table 8, the CRs and PSNRs of images which have not been processed by color conversion are smaller than those of images from converting RGB to YCrCb. In addition, for residue coding methods, the PSNRs of these four methods are almost the same when the BER is 0.001. Bi-level block coding algorithm has a higher PSNR than other two algorithms when the BER is 0.005. Even though the PSNR from 1-D bi-level block coding has almost the same PSNR when comparing with the PSNR from 2-D bi-level block coding, the 2-D bi-level block coding method achieves a higher CR.

Fig. 7 shows the results for BER at 0.005. Note that only the 2-D bi-level block coding results are included in Fig. 7 since 1-D bi-level block coding has almost the same PNSR value. When the BER increases, there exists more significant impact on the image quality, resulting in image distortion.

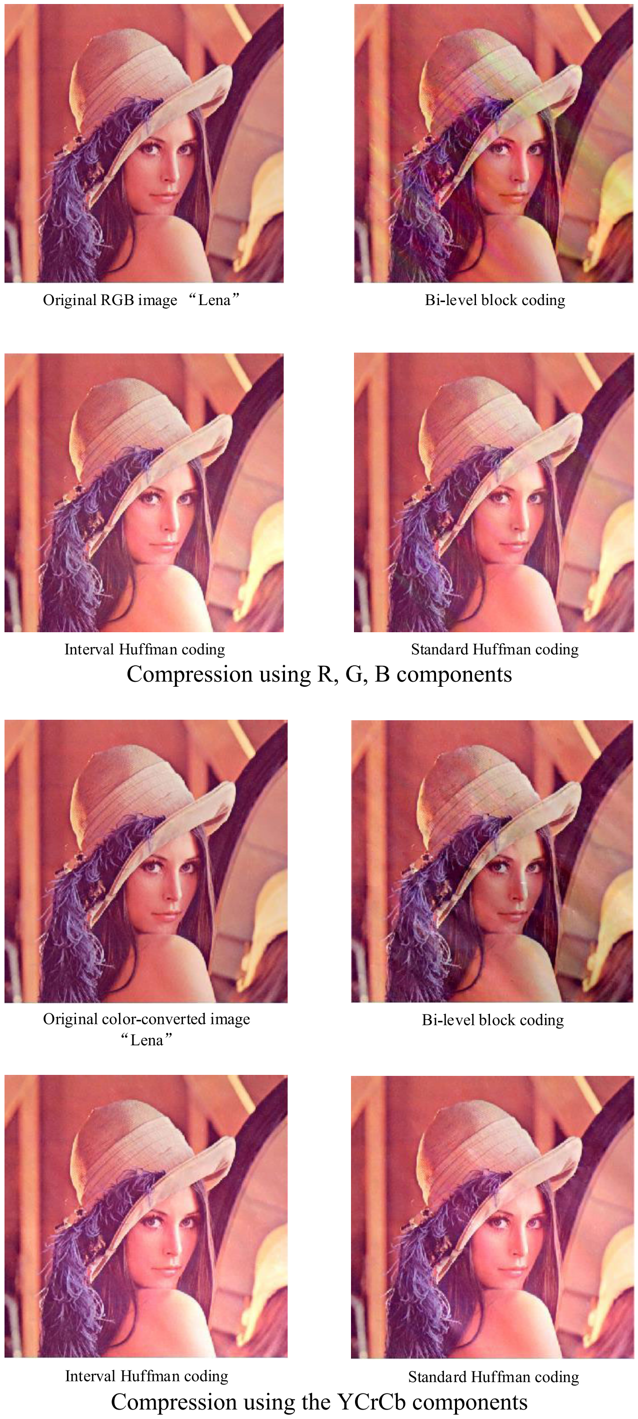
Figs. 8-11 show the PSNR performances with the change of the bit-error rate for color-converted "Lena", "Baboon", "Pepper", and "Airplane" images in the YCrCb color space, respectively. Each PSNR in each figure is obtained by averaging the values from 10 independent runs.

It can be seen that the PSNRs from 2-D and 1-D bi-level block coding algorithms are close. When the BER is more than 0.001, bi-level block coding offers a higher value of PSNR than the other three methods. On the other hand, the "Lena" image using bi-level block coding has the lowest value of PSNR when the BER is less than 0.001. However, bi-level block coding algorithm still maintains an excellent image quality.

In order to compare CR performances, we list the results from using four different residue coding methods in Table 9. As shown in Table 9, two industry standard lossless compression algorithms of JPEG2000 [35] and WebP [36] are used to create reference values by adding (7,4) Hamming codes so that they are more robust to bit errors for fair comparisons.

As shown in Table 9, the bi-level block coding algorithms achieve higher CRs. The new 2-D bi-level block coding algorithm has a higher CR than the 1-D bi-level block coding algorithm. In terms of both bit-error resilience and compression ratio, the 2-D bi-level block coding algorithm with prediction in the YCrCb color space offers the best performance in general.

Clearly, it can be seen that the proposed YCrCb predictive 2-D bi-level block coding has the following advantages: (1) it offers a highest CR in comparison with the other different combinations, since the first efficiency is achieved via the best selection of the RGB to YCrCb conversion and


Original RGB image “Lena”

Bi-level block coding

Interval Huffman coding

Standard Huffman coding

Compression using R, G, B components

Original color-converted image “Lena”

Bi-level block coding

Interval Huffman coding

Standard Huffman coding

Compression using the YCrCb components

FIGURE 7. Comparison results using predictive 2-D bi-level block coding, predictive interval Huffman coding and predictive Huffman coding at BER = 0.005 for “Lena” image.

prediction; and the second efficiency is further obtained via 2-D bi-level block coding; (2) for a noisy environment with a larger bit-error rate, that is, 0.001 < BER <0.01, the proposed method restores the best color image quality, that is, achieving the highest PSNR value. Many additional simulations show that the PNSR value using prediction in the YCrCb color space is about 2∼3 dB higher than the one from the prediction in the RGB color space. Note that our method has a significant advantage for compressing color images. For the gray-scale images, the two-stage method in [27] can be easily applied. Our future work will include the measurement comparisons

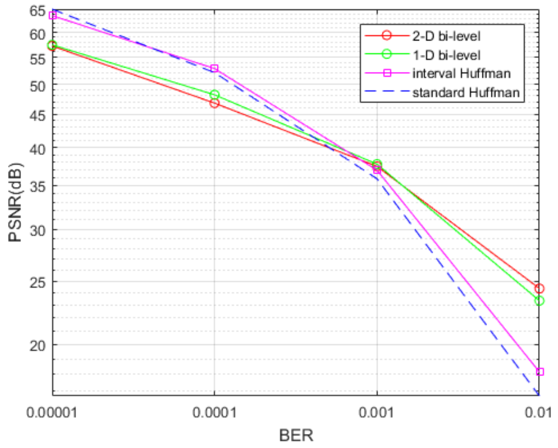


FIGURE 8. PSNR performances versus the bit-error rate for “Lena” image.

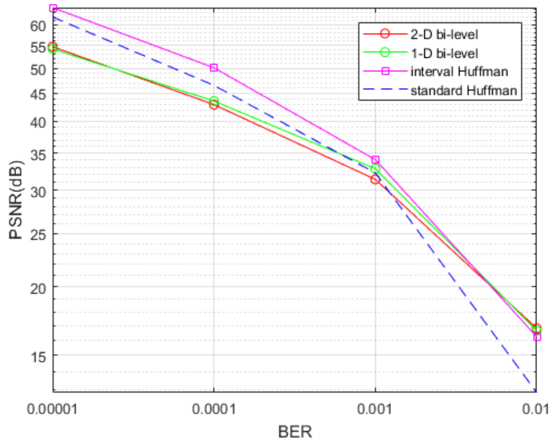


FIGURE 9. PSNR performances versus the bit-error rate for “Baboon” image.

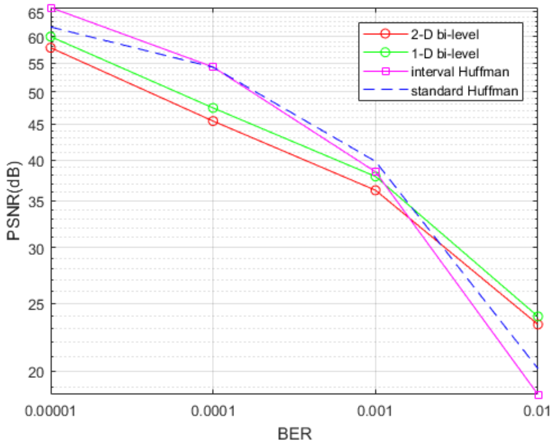


FIGURE 10. PSNR performances versus the bit-error rate for “Pepper” image.

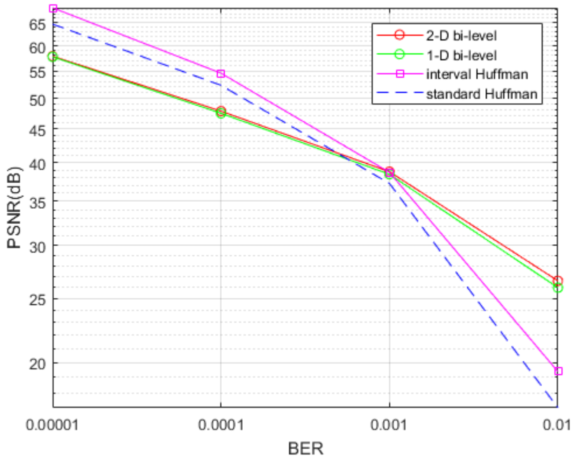


FIGURE 11. PSNR performances versus the bit-error rate for "Airplane" image.

TABLE 9. Compression ratio (CR) performances versus different residue methods and industry standards.

Algorithm	Lena	Baboon	Pepper	Airplane
2-D block coding YCrCb	**1.4456**	**1.1647**	**1.3227**	**1.5104**
2-D block coding RGB	1.33760	1.1117	1.3115	1.4559
1-D block coding YCrCb	1.4276	1.1540	1.3104	1.4929
1-D block coding RGB	1.3490	1.0020	1.2923	1.4349
Interval Huffman YCrCb	1.2375	1.0265	1.1294	1.2079
Interval Huffman RGB	1.1639	0.9785	1.0980	1.2080
Standard Huffman YCrCb	1.0735	0.8927	1.0098	1.1667
Standard Huffman RGB	1.0532	0.8616	1.0141	1.1359
JPEG 2000 (7,4)/ JPEG 2000	1.0068/ 1.7620	0.7585/ 1.3274	0.9267/ 1.6218	1.1888/ 2.0804
WebP (7,4) /WebP	1.0491/ 1.8359	0.7756/ 1.3514	0.9708/ 1.6988	1.1889/ 2.0808

of image quality in the bit-error environment using the PNSR as well as a similar index such as the structural similarity (SSIM).

VII. CONCLUSION

In this paper, we have first developed a general bi-level block coding method so that a 2-D bi-level block coding can be proposed for image residue compression. Next, we have proposed a three-stage compression framework for color image compression with bit-error awareness. It consists of converting the RGB image to the YCrCb image; applying linear predictors for the YCrCb components; encoding residue sequence and packing key parameters protected using (7,4) Hamming codes. A particle swarm optimization (PSO) algorithm is also applied to determine the best combination of reversible color transformations from the RGB color space to the YCrCb color space, and image predictors.

Our experimental results validate the coding performances in terms of the compression ratio (CR) and peak signal to noise ratio (PSNR). When the bit-error rate (BER) is larger than 0.001, the bi-level block coding algorithm offers the highest PSNR. Even if the BER is less than 0.001, this method still maintains a good value of PSNR. In terms of compression ratio, the new 2-D bi-level block coding algorithm achieves the highest CR.

REFERENCES

[1] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," *IEEE Trans. Commun.*, vol. 45, no. 4, pp. 437–444, Apr. 1997.

[2] T. Lin and P. Hao, "Compound image compression for real-time computer screen image transmission," *IEEE Trans. Image Process.*, vol. 14, no. 8, pp. 993–1005, Aug. 2005.

[3] L. Shen and R. M. Rangayyan, "A segmentation-based lossless image coding method for high-resolution medical image compression," *IEEE Trans. Med. Imag.*, vol. 16, no. 3, pp. 301–307, Jun. 1997.

[4] J. Mielikainen and B. Huang, "Lossless compression of hyperspectral images using clustered linear prediction with adaptive prediction length," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 6, pp. 1118–1121, Nov. 2012.

[5] S.-G. Miaou, F.-S. Ke, and S.-C. Chen, "A lossless compression method for medical image sequences using JPEG-LS and interframe coding," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 5, pp. 818–821, Sep. 2009.

[6] V. Sanchez, R. Abugharbieh, and P. Nasiopoulos, "Symmetry-based scalable lossless compression of 3D medical image data," *IEEE Trans. Med. Imag.*, vol. 28, no. 7, pp. 1062–1072, Jul. 2009.

[7] H. Wu, X. Sun, J. Yang, W. Zeng, and F. Wu, "Lossless compression of JPEG coded photo collections," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2684–2696, Jun. 2016.

[8] S. Kim and N. I. Cho, "Hierarchical prediction and context adaptive coding for lossless color image compression," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 445–449, Jan. 2014.

[9] R. Kannan and C. Eswaran, "Lossless compression schemes for ECG signals using neural network predictors," *EURASIP J. Adv. Signal Process.*, vol. 2007, no. 1, pp. 1–20, Dec. 2007.

[10] R. Starosolski, "Simple fast and adaptive lossless image compression algorithm," *Softw., Pract. Exp.*, vol. 37, no. 1, pp. 65–91, Jan. 2007.

[11] T. Leung, M. W. Marcellin, and A. Bilgin, "Visually lossless compression of windowed images," in *Proc. Data Compress. Conf.*, Mar. 2013, p. 504.

[12] A. Koski, "Lossless ECG encoding," *Comput. Methods Programs Biomed.*, vol. 52, no. 1, pp. 23–33, 1997.

[13] V. Sanchez, P. Nasiopoulos, and R. Abugharbieh, "Efficient lossless compression of 4-D medical images based on the advanced video coding scheme," *IEEE Trans. Inf. Technol. Biomed.*, vol. 12, no. 4, pp. 442–446, Jul. 2008.

[14] S. D. Stearns, L. Tan, and N. Magotra, "A bi-level coding technique for compressing broadband residue sequences," *Digit. Signal Process.*, vol. 2, no. 3, pp. 146–156, Jul. 1992.

[15] S. D. Stearns, "Arithmetic coding in lossless waveform compression," *IEEE Trans. Signal Process.*, vol. 43, no. 8, pp. 1874–1879, Aug. 1995.

[16] S. D. Stearns, *Digital Signal Processing With Examples in MATLAB*. Boca Raton, FL, USA: CRC Press, 2002.

[17] L. Tan and J. Jiang, *Digital Signal Processing: Fundamentals and Applications*, 3rd ed. Amsterdam, The Netherlands: Elsevier, 2018.

[18] N. Sriraam and C. Eswaran, "Context based error modeling for lossless compression of EEG signals using neural networks," *J. Med. Syst.*, vol. 30, no. 6, pp. 439–448, Nov. 2006.

[19] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS," *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1309–1324, Aug. 2000.

[20] S. D. Stearns, L.-Z. Tan, and N. Magotra, "Lossless compression of waveform data for efficient storage and transmission," *IEEE Trans. Geosci. Remote Sens.*, vol. 31, no. 3, pp. 645–654, May 1993.

[21] J. Ding, I. Wang, and H.-Y. Chen, "Improved efficiency on adaptive arithmetic coding for data compression using range-adjusting scheme, increasingly adjusting step, and mutual-learning scheme," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 12, pp. 3412–3423, Dec. 2018.

[22] A. K. Shukla, A. Singh, B. Singh, and A. Kumar, "A secure and high-capacity data-hiding method using compression, encryption and optimized pixel value differencing," *IEEE Access*, vol. 6, pp. 51130–51139, Sep. 2018.

[23] Z. Chen, K. Fan, S. Wang, L. Duan, W. Lin, and A. C. Kot, "Toward intelligent sensing: Intermediate deep feature compression," *IEEE Trans. Image Process.*, vol. 29, pp. 2230–2243, 2020.

[24] T. Strutz and A. Leipnitz, "Reversible color spaces without increased bit depth and their adaptive selection," *IEEE Signal Process. Lett.*, vol. 22, no. 9, pp. 1269–1273, Sep. 2015.

[25] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1983.

[26] S. Rhee, C. Kim, J. Kim, and Y. Jee, "Concatenated Reed–Solomon code with Hamming code for DRAM controller," in *Proc. 2nd Int. Conf. Comput. Eng. Appl.*, Bali, Island, 2010, pp. 291–295.

[27] L. Tan and L. Wang, "Bit-error aware lossless image compression," *Int. J. Mod. Eng.*, vol. 11, no. 2, pp. 54–59, 2011.

[28] G. Zeng and N. Ahmed, "A block coding technique for encoding sparse binary patterns," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 5, pp. 778–780, May 1989.

[29] L. Tan and J. Jiang, "A bi-level block coding technique for encoding data sequences with sparse distributions," *Technol. Interface J.*, vol. 9, no. 1, 2008.

[30] L. Tan, J. Jiang, and Y. Zhang, "Bit-error aware lossless compression of waveform data," *IEEE Signal Process. Lett.*, vol. 17, no. 6, pp. 547–550, Jun. 2010.

[31] F. Marcelloni and M. Vecchio, "A simple algorithm for data compression in wireless sensor networks," *IEEE Commun. Lett.*, vol. 12, no. 6, pp. 411–413, Jun. 2008.

[32] X. Peng, J. Hou, L. Tan, J. Chen, J. Jiang, and X. Guo, "Bit-error aware lossless color image compression," in *Proc. IEEE Int. Conf. Electro Inf. Technol. (EIT)*, Brookings, South Dakota, May 2019, pp. 126–131.

[33] M. Ahmadieh Khanesar, M. Teshnehlab, and M. Aliyari Shoorehdeli, "A novel binary particle swarm optimization," in *Proc. Medit. Conf. Control Autom.*, Athens, Greece, Jun. 2007, pp. 1–6.

[34] X. Wu, "A density adjustment based particle swarm optimization learning algorithm for neural network design," in *Proc. Int. Conf. Electr. Control Eng.*, Yichang, China, Sep. 2011, pp. 2829–2832.

[35] (May 16, 2020). *OpenJPEG*. [Online]. Available: https://www.openjpeg.org/

[36] (May 16, 2020). *A New Image Format For the Web*. [Online]. Available: https://developers.google.com/speed/webp

XUAN PENG received the B.S. degree in materials science and engineering from the Beijing Institute of Technology, Beijing, China, in 2017, and the M.S. degree in electrical and computer engineering, from Purdue University Northwest, Hammond, IN, USA, in 2019. His research interests include digital signal processing, data and image compression, and image processing.

JEAN JIANG (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Southeast University, Nanjing, China, in 1982 and 1985, respectively, and the Ph.D. degree in electrical engineering from The University of New Mexico, Albuquerque, NM, USA, in 1992.

She is currently an Associate Professor with the College of Technology, Purdue University Northwest, Hammond, IN, USA. She has authored or coauthored two textbooks *Analog Signal Processing and Filter Design* (Linus Publications, Second Edition, 2016) and *Digital Signal Processing: Fundamentals and Applications* (Elsevier, Third Edition, 2018). Her research interests include digital signal processing, adaptive signal processing, control systems, computer vision, and robotics.

LIZHE TAN (Senior Member, IEEE) received the B.S. degree from Southeast University, Nanjing, China, in 1984, and the M.S. degree in engineering mechanics and the M.S. and Ph.D. degrees in electrical engineering from The University of New Mexico, Albuquerque, NM, USA, in 1987, 1989, and 1992, respectively.

He is currently a Professor with the Department of Electrical and Computer Engineering, Purdue University Northwest, Hammond, IN, USA. He has authored or coauthored two textbooks *Analog Signal Processing and Filter Design* (Linus Publications, Second Edition, 2016) and *Digital Signal Processing: Fundamentals and Applications* (Elsevier, Third Edition, 2018). He holds a granted U.S. patent. His research interests include digital signal processing, adaptive signal processing, control systems, computer vision, and robotics. He has served as an Associate Editor for the *International Journal of Engineering Research and Innovation*.

JINTAO HOU received the B.S. degree in measurement and control technology from the North University of China, Taiyuan, China, in 2017, and the M.S. degree in electrical and computer engineering from Purdue University Northwest, Hammond, IN, USA, in 2020. His research interests include digital signal processing, image processing, and computer vision.