Journal of
Electronic Imaging

JElectronicImaging.org

Technique for lossless compression of color images based on hierarchical prediction, inversion, and context adaptive coding

Basar Koc
Ziya Arnavut
Dilip Sarkar
Hüseyin Koçak

SPIE. IS&T imaging.org

Basar Koc, Ziya Arnavut, Dilip Sarkar, Hüseyin Koçak, “Technique for lossless compression of color images based on hierarchical prediction, inversion, and context adaptive coding,” *J. Electron. Imaging* **28**(5), 053007 (2019), doi: 10.1117/1.JEI.28.5.053007.

Technique for lossless compression of color images based on hierarchical prediction, inversion, and context adaptive coding

Basar Koc,[a] **Ziya Arnavut,**[b,*] **Dilip Sarkar,**[c] **and Hüseyin Koçak**[c]
[a]Stetson University, Department of Computer Science, DeLand, Florida, United States
[b]State University of New York at Fredonia, Department of Computer and Information Sciences, Fredonia, New York, United States
[c]University of Miami, Department of Computer Science, Coral Gables, Florida, United States

Abstract. Among the variety of multimedia formats, color images play a prominent role. A technique for lossless compression of color images is introduced. The technique is composed of first transforming a red, green, and blue image into luminance and chrominance domain (YC_uC_v). Then, the luminance channel Y is compressed with a context-based, adaptive, lossless image coding technique (CALIC). After processing the chrominance channels with a hierarchical prediction technique that was introduced earlier, Burrows–Wheeler inversion coder or JPEG 2000 is used in compression of those C_u and C_v channels. It is demonstrated that, on a wide variety of images, particularly on medical images, the technique achieves substantial compression gains over other well-known compression schemes, such as CALIC, M-CALIC, Better Portable Graphics, JPEG-LS, JPEG 2000, and the previously proposed hierarchical prediction and context-adaptive coding technique LCIC. [DOI: 10.1117/1.JEI.28.5.053007]

Keywords: lossless color image compression; inversion coding; context-based, adaptive, lossless image coding; JPEG 2000; hierarchical decomposition and prediction; Burrows–Wheeler transformation.

Paper 190396 received May 2, 2019; accepted for publication Aug. 27, 2019; published online Sep. 16, 2019.

1 Introduction

The growth of data-intensive multimedia-based web applications and advancements in image sensors and processing units of mobile devices have made compression of multimedia content central to storage and communication technology. Efficient data and image compression algorithms are therefore required to minimize the cost and demands especially in devices and networks with limited resources. There are two main ways, lossy and lossless, to compress data. The lossy compression strategy, as the name implies, involves some loss of information, i.e., data that have been compressed by a lossy technique cannot be exactly reproduced through decompression. Depending on the application, lossy techniques can be used to increase the channel throughput. For example, images viewed on television have significant loss of information with respect to the original images, but this does not cause a problem since the human visual system is not perfect. Good lossy compression techniques can achieve high compression ratios on typical real-life images and videos.

In lossless compression, while the original quality of an image is preserved, the reduction in the file size may not be as good as in lossy compression. A few applications of lossless compression are text, encoded text for the World Wide Web, such as HTML, XML, and SHTML documents, computer-generated data (executable code), and images. In some applications, such as medical imaging where the quality of an image is the highest priority and any loss of data is not tolerable, lossless compression is preferred.[2–8] The purpose of this paper is to introduce a lossless compression algorithm for color images.

In most color images, each pixel is represented by 24-bits, or 8-bits per channels, red, green, and blue (RGB). To reduce interchannel correlations and maximize compression gain, the majority of image compression algorithms, when used on color images, utilize a variety of reversible color transformations (RCT) on RGB channels, as described in, for example, Refs. 9–14. Commonly used color transformations map RGB channels into luminance Y and chrominance C_u and C_v channels. The luminance channel Y is essentially a grayscale version of the original image. Indeed, the origins of this transformation date back to the early days of television to ensure good reception of color transmissions on black-and-white television sets. In lossless color image compression, care must be exercised in the choice of RCT, because some color transformations may not be truly invertible due to loss of precision in noninteger arithmetic operations in direct and inverse transforms.[10] The widely used lossless color image compressor JPEG 2000[15] uses an invertible RCT.

Kim and Cho[16] proposed a technique for lossless compression of color images. In this technique, an RCT is first applied to a color image. The luminance channel Y is compressed with JPEG 2000. The chrominance channels C_u and C_v are divided into two subimages as odd and even rows. To improve prediction accuracy and estimation of error, the odd subimage X_o is predicted using the even subimage X_e. The even subimage X_e is compressed with JPEG 2000 and stored for decoding. The odd subimage X_o is encoded with an arithmetic coder after applying a context modeling for the prediction errors obtained from neighboring pixels.

*Address all correspondence to Ziya Arnavut, E-mail: ziya.arnavut@fredonia.edu

A preliminary version of this manuscript was presented as a poster at 2019 IEEE Data Compression Conference (DCC)

1017-9909/2019/$28.00 © 2019 SPIE and IS&T

Kim and Cho[17] improved their previous algorithm presented in Ref. 16 by using a different RCT similar to the one in Ref. 11 and developed a more advanced hierarchical prediction technique, which they called LCIC. They demonstrated in the same publication that on a variety of standard color image sets, LCIC outperformed JPEG 2000 and JPEG-XR.[18]

In this paper, we propose a lossless color image compression technique, called BWIC_I, in part based on the framework of LCIC. In BWIC_I, first, the RGB channels are converted into YC_uC_v channels with an RCT. We compressed the Y channels of standard test image sets with a multitude of algorithms and determined that the context-based, adaptive, lossless image coding (CALIC) algorithm outperformed, on average, the other compressors, including JPEG 2000. Therefore, in BWIC_I, the Y channel is compressed with CALIC. Second, using the hierarchical prediction technique from LCIC, the chrominance channels are reduced to a quarter of their original sizes. These preprocessed channels can have rather different characteristics. Therefore, employing an appropriate compressor on each preprocessed chrominance channel can improve the overall compression gain. Our experiments have shown that JPEG 2000 and Burrows–Wheeler inversion coder (BWIC) are the two best performers for this task. Hence, BWIC_I compresses the quarter-sized preprocessed C_u and C_v and chooses the best compressor for each channel.

Compression gains of BWIC_I over JPEG 2000 are 10.7% on Kodak, 28.4% on Medical, and 8.6% on commercial digital camera image sets. Moreover, these compression gains of the proposed technique are on average a 4.5% improvement over the LCIC gains reported in Ref. 17, while on medical image set, gains are over 11.1%. Admittedly, our lossless color image compression algorithm BWIC_I is a carefully designed ensemble of existing ideas and algorithms. However, as the performance data above quantifies, our algorithm decidedly outperforms the existing methods on many standard test color image sets.

This paper is organized as follows: In Sec. 2, we briefly describe each technique used in our proposed coding scheme. In Sec. 3, the proposed technique, along with RCT, hierarchical prediction, and context-adaptive coding techniques, is described. In Sec. 4, the experimental results are discussed and shown that our proposed technique, on a wide variety of image sets particularly on medical images, achieves better compression results than the well-known standard compression schemes JPEG-LS, JPEG 2000, Better Portable Graphics (BPG), and other previously proposed techniques, such as LCIC. Finally, Sec. 5 gives a summary of the conclusions of the present work.

2 Related Work

In this section, we briefly describe each of the methods utilized to set the stage for discussion of the proposed coding technique for lossless compression of color images.

2.1 *Context-Based, Adaptive, Lossless Image Coding*

CALIC, proposed by Wu and Memon,[19] was one of the algorithms considered in the process of JPEG standardization. It was designed to be a lossless image codec but CALIC was not selected as the JPEG standard due to its complexity, as stated by the authors:[19] "both the current lossless JPEG and FELICS are rather simple techniques that require minimal memory and computation resources. CALIC, on the other hand, is more complex and does require more resources, although the increase in memory and computation resources is modest."

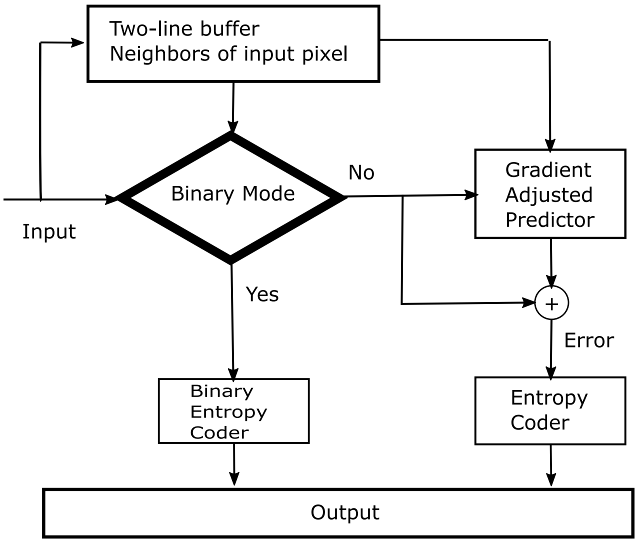


Fig. 1 CALIC encoding algorithm.

CALIC works in raster scan order and uses only two previous scan lines to predict the current pixel. To obtain higher compression gain, CALIC uses two different compression modes, binary and continuous tone, based on the previously encoded pixel information. The encoding algorithm switches between compression modes. When the previous two lines have fewer than three different values, the binary encoding mode is used in encoding the current pixel. This mode improves compression gain on images with uniform image areas and uses pixel values instead of prediction values. For other cases, continuous mode is utilized by the system. Continuous tone encoding mode consists of multiple components: the gradient-adjusted predictor (GAP), quantization, the context modeling, and the entropy coder. GAP is a nonlinear predictor that adjusts coefficient prediction based on the intensity of neighboring pixels. It uses seven neighboring pixels. The output from GAP is used in the quantization of the error energy estimator and in encoding after calculating the error value as the difference between estimated and predicted pixel values. The performance of the GAP is improved further using context modeling. A schematic description of CALIC's encoder is shown in Fig. 1. For further details about CALIC, the interested reader is referred to Ref. 19.

CALIC was originally designed for compression of 8-bit grayscale images. Wu and Memon[20] later developed an interband version of CALIC suitable for effective compression of multispectral images like color and remotely sensed images. More recently, Magli et al.[21] proposed a compressor for hyperspectral images, called M-CALIC, on top of an optimized CALIC-based compression engine. M-CALIC outperforms interband CALIC on representative test images.[21]

2.2 *JPEG 2000*

JPEG 2000 is an ISO/IEC still image compression standard based on discrete wavelet transform. It was designed to

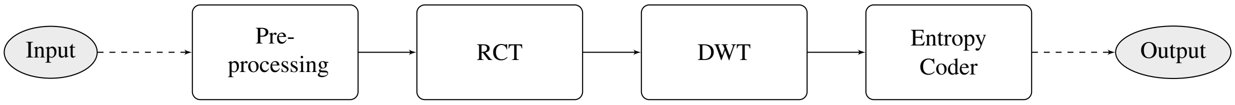


Fig. 2 The steps in JPEG 2000 lossless mode.

replace the DCT-based original JPEG image compression algorithm by not only improving the compression rate but also incorporating features such as scalability and editability. In lossless mode, JPEG 2000 uses a RCT with a Cohen–Daubechies–Feauveau biorthogonal 5/3 integer wavelet transform.

After the transformation, the lossless mode does not quantize the values. For the entropy coder phase, original JPEG utilizes a Huffman coder, whereas JPEG 2000 uses a context-based adaptive binary arithmetic coder (BAC). A schematic description of JPEG 2000 is given in Fig. 2. For further details about JPEG 2000, the interested reader is referred to Refs. 15 and 22–24.

2.3 *BWIC*

One of the most prominent transformation-based algorithms used in data compression is the invertible Burrows–Wheeler transformation (BWT), a lexical-sorting based transformation.[25,26] Since the introduction of BWT, several authors have proposed improvements to the original algorithm. Schindler[27] suggested a different block-sorting technique and developed the szip coder. Arnavut[28] introduced the use of inversion coding (IC) to block-sort BWT and presented a faster and more memory-efficient lossless data compression algorithm and called it BWIC. When BWIC is applied to text or image data, it achieves, on average, better compression rates than Lempel–Ziv–Welch techniques.[28]

BWIC consists of four basic steps: lexical sorting transformation, inversion coder, followed by run-length and entropy coders. A schematic description of BWIC is displayed in Fig. 3. The effects of BWT with IC on the pixel values of an image file are shown in the histograms in Fig. 4.

The first step in BWIC is BWT. The main goal of the BWT is to collect symbols in lexically similar contexts near to each other. It is important to note that BWT does not compress data. The time complexity of the BWT is $\mathcal{O}(n \log n)$. It is known to be a very effective transformation for text data. Compression results of BWT-based coders are highly related to the window (block) size used in the transformation. While the selection of bigger window sizes helps to obtain better compression, it requires more memory space and also reduces the speed. For the results reported in this paper, the maximum window (block) size is set to 8 MB. After trying several different block sizes, we experimentally determined that good compression can be obtained with 8 MB block size without sacrificing speed. Hence, files bigger than 8 MB are processed in chunks of 8 MB.

In the second step, BWIC uses IC, also called inversion ranks or inversion frequencies.[28] Most variants of BWT based compressors, such as bzip2 and szip, employ a move-to-front (MTF) transformation. MTF was introduced by Bentley et al.[29] (also independently discovered by Elias[30]). MTF keeps the most current symbols at the beginning of an ordered list and needs $\mathcal{O}(nm)$ time. In Refs. 28 and 31, IC and MTF were investigated. It was shown that, when used in the second step of a Burrows–Wheeler based compression scheme, IC yields better compressible sequences on more data types than MTF. IC forms a sequence I of integers from the range of $[0, n-1]$ over an alphabet A. For each character $a_j \in A$, the algorithm scans a given sequence S and

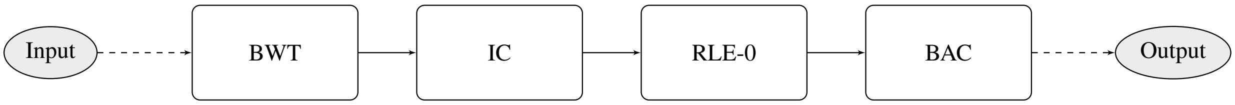


Fig. 3 The steps in BWIC.

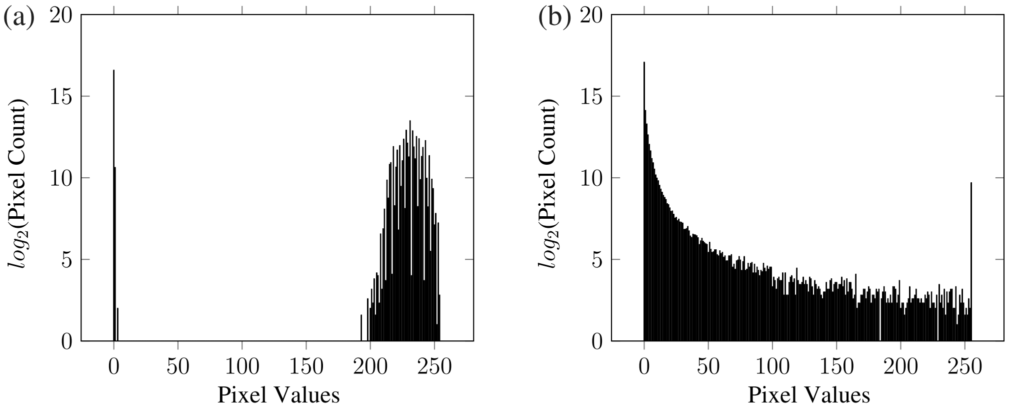


Fig. 4 The cumulative effects of BWT and IC are to reduce the number of large values in a data set, as seen in the chrominance channel C_v of image Kodak01. (a) Histogram of pixel values before processing and (b) after processing with BWT and IC.

whenever it discovers the first occurrence of the character a_j, it outputs its position in S. After that, it outputs the number of characters greater than a_j that occurred since the last identification of the a_j. To obtain the original data, in addition to the inversion rank sequence I, frequency f_j of each a_j in S needs to be transmitted to the receiver. The interested reader is referred to Refs. 28 and 31 for further information.

The third step is a special run-length encoding called the zero run-length coder (RLE-0). Zero is a dominant symbol of the sequences processed first by BWT and later by IC when applied to commonly used data types such as text, image, audio, and video. Thus, there are many long runs in sequences processed by IC consisting of zeros, which are called 0-runs. This may cause difficulties in probability estimations when encoding such sequences. To overcome this problem, a transform that treats 0-runs in a special way can be used. Experimental results show that using an RLE-0 transform improves the compression ratio. A detailed explanation of RLE-0 can be found in Refs. 32 and 33.

The last step of BWIC is an entropy coder. Analysis of probability distributions of IC shows that zero is a dominant symbol, and the probability of higher symbols usually decreases monotonically. Fenwick, who studied the sequences generated by the MTF coder after BWT has been applied, developed a hierarchical model[32] in which symbols are divided into some classes, where within a class the differences in probabilities of occurrence of symbols are relatively small. He has shown that for the best compression, it is necessary to use an adaptive arithmetic coder.[34] In our implementation of BWIC, we have chosen a BAC, as described by Deorowicz in Ref. 35. Overall, BWIC completes the entire compression process in $\mathcal{O}(n \log n)$ time.

3 Proposed Coding Scheme BWIC_I

The overall structure of our proposed lossless compression scheme for color images is depicted in Fig. 5.

As the first step in the proposed coding process, the RGB channels of an image are transformed into the luminance channel Y, and the chrominance channels, C_u and C_v, using the following invertible RCT:[11]

$$\begin{aligned}\text{Forward transform: } C_u &= B - \text{round}\left\{\frac{87}{256}R + \frac{169}{256}G\right\},\\ C_v &= R - G,\\ Y &= G + \text{round}\left\{\frac{86}{256}C_v + \frac{29}{256}C_u\right\}.\end{aligned} \tag{1}$$

$$\begin{aligned}\text{Inverse transform: } G &= Y - \text{round}\left\{\frac{86}{256}C_v + \frac{29}{256}C_u\right\},\\ R &= C_v + G,\\ B &= C_u + \text{round}\left\{\frac{87}{256}R + \frac{169}{256}G\right\},\end{aligned} \tag{2}$$

where the floating-point numbers are rounded to the nearest integer.

The luminance channel Y is a grayscale version of the image. After applying an RCT transformation, Kim and Cho[17] used JPEG 2000 to compress Y. We tried several coding techniques, such as CALIC, JPEG 2000, JPEG-LS,[36] BWIC, and concluded that CALIC yields the best compression. Hence, we used CALIC to compress Y.

Before compression, the chrominance channels C_u and C_v are preprocessed with a hierarchical decomposition scheme proposed by Kim and Cho.[16,17] (This scheme is similar to the one proposed by Roman in Ref. 37.) Pixels of a chrominance channel are first decomposed into two subimages: X_o consisting of the odd rows and X_e consisting of the even rows. For the compression of X_o, X_e are used in the prediction algorithm. While X_e are stored, X_o are predicted using X_e, and error values are encoded using an arithmetic coder. The subimage X_e is further decomposed into odd and even columns using the same approach. To reduce large prediction errors near edges, a directional prediction technique is utilized. For each pixel $x_o(i, j)$ in X_o, the horizontal (left neighbor) and the vertical (average of upper and lower neighbors) predictors are calculated:

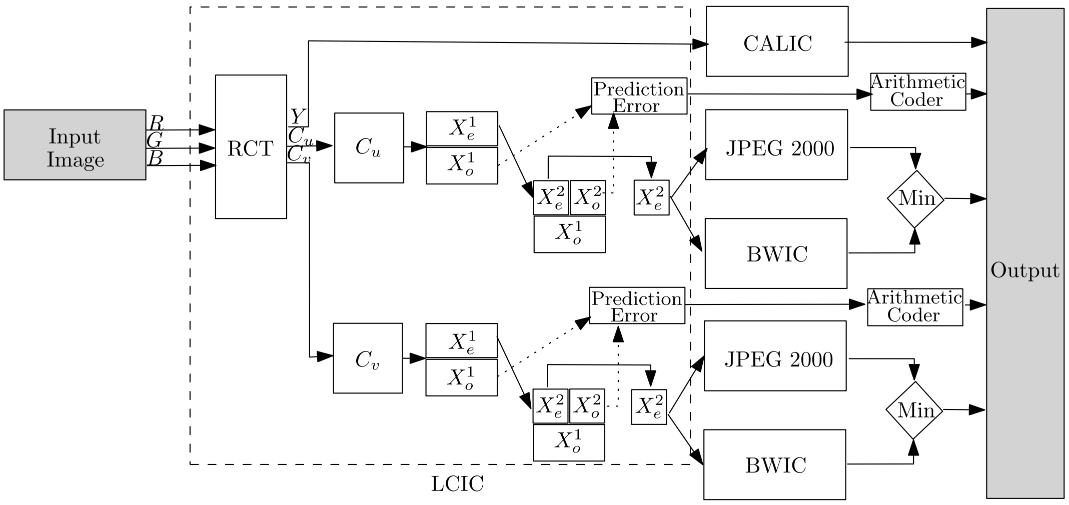


Fig. 5 Proposed lossless compression scheme BWIC_I.

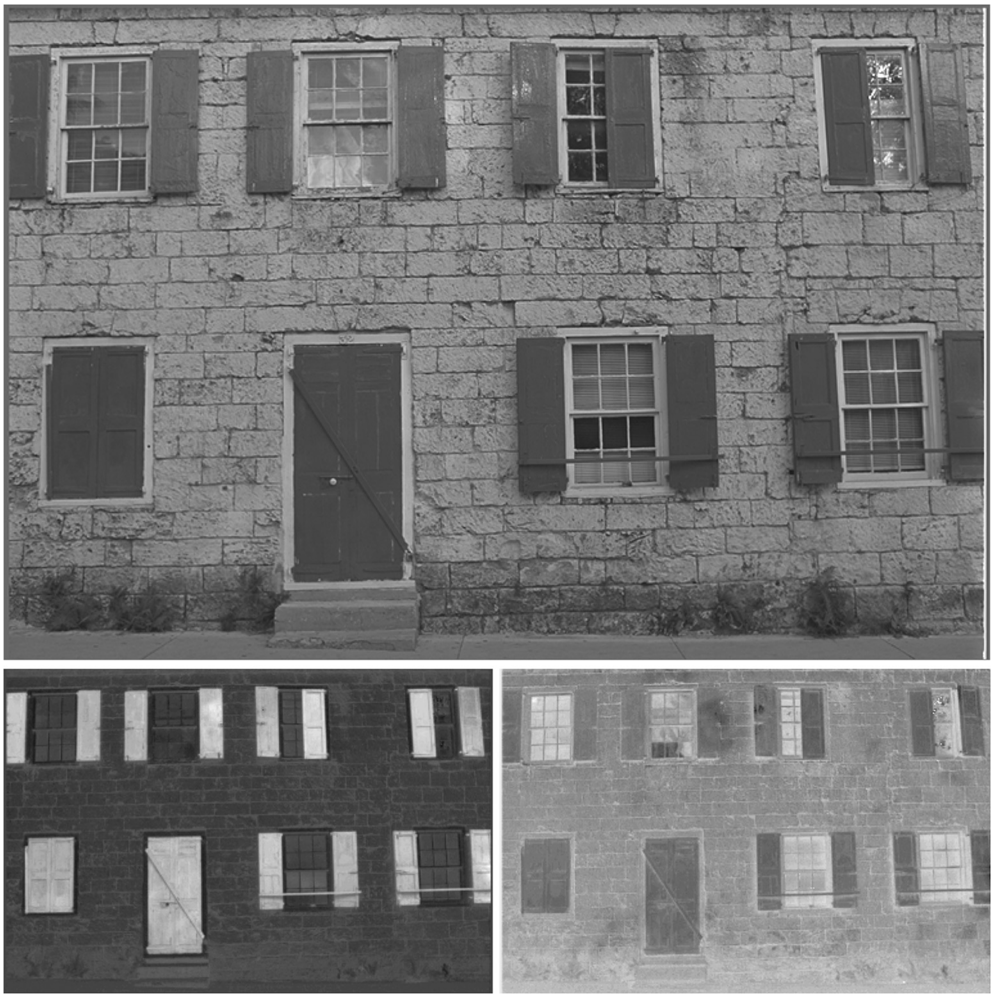


Fig. 6 The luminance Y and the preprocessed chrominance C_u and C_v channels with the hierarchical decomposition scheme of Kodak01 from the Kodak image set. (The MATLAB function mat2gray() that converts a matrix to grayscale image is used to make the negative pixel values visible.)

$$\hat{x}_h(i,j) = x_o(i,j-1)$$

$$\hat{x}_v(i,j) = \text{round}\left(\frac{x_e(i,j)+x_e(i+1,j)}{2}\right). \qquad (3)$$

Based on the values of $|x_o(i,j)-\hat{x}_h(i,j)|$ and $|x_o(i,j)-\hat{x}_v(i,j)|$, a mode selection is made; the vertical predictor is used for most pixels, and the horizontal predictor is used when there is a “strong” horizontal edge. The details of the hierarchical decomposition and pixel prediction processes are presented in Algorithms 1 and 2 of Ref. 17.

Although the prediction method summarized above produces small errors, near the edges and in highly textured regions, they can be relatively large. To further reduce such prediction error values, statistical properties of prediction errors are used and error values are quantized based on local activity, as stated by the authors of Ref. 17:

> We model the prediction error as a random variable with pdf $P(e|C_n)$, where C_n is the coding context that reflects the magnitude of edges and textures. Specifically, C_n is the level of quantization steps of pixel activity $\sigma(i,j)$ defined as follows:
>
> $$\sigma(i,j) = |x_e(i,j) - x_e(i+1,j)|.$$
>
> Note that the local activity and its quantization steps are calculated with the pixels in X_e, because all the pixels of X_e are available and its statistical property would be almost the same as that of X_o.

As a result of this hierarchical preprocessing, the size of each chrominance channel is reduced to a quarter of its original size. In Fig. 6, the luminance Y channel and the preprocessed chrominance channels C_u and C_v with the hierarchical decomposition scheme of an image from the Kodak set (Kodak01) are shown.

The hierarchical decomposition process is reversible for images with an even number of rows and columns, and the complete chrominance channels can be recovered without loss of information in decoding. To ensure the reversibility of the hierarchical decomposition scheme for images with an odd number of rows or columns, we temporarily apply appropriate padding to such images during compression and decompression.

Kim and Cho[17] used JPEG 2000 to compress the preprocessed chrominance channels as well. It is possible to achieve better compression gains on these channels. Indeed, the distribution of pixel values in C_u and C_v can differ considerably, as evident in Fig. 7, where the histograms of

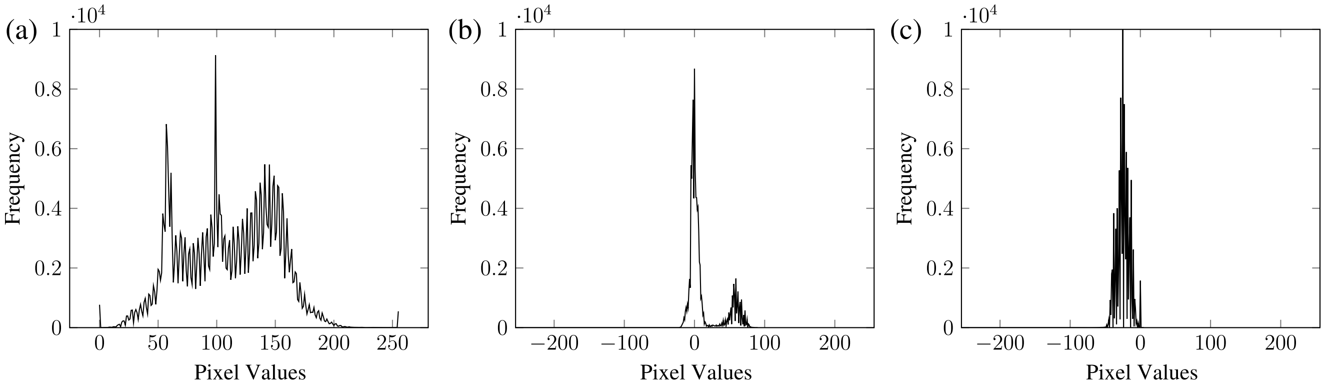


Fig. 7 (a)–(c) Histograms of pixel values of Y and the preprocessed C_u and C_v channels of images from Fig. 6.

these channels for the Kodak01 image in Fig. 6 are plotted. Therefore, it behooves one to employ different compression algorithms appropriate for each chrominance channel. To this end, we experimented with various compression algorithms (including CALIC, M-CALIC, PNG, JPEG-LS, BPG, JPEG 2000, BWIC) on standard image sets and decided that JPEG 2000 and BWIC were the two top performers. However, for a given particular image, despite trying various statistical sampling techniques, it does not seem possible to determine *a priori* which of the two compressors will prevail. While on some images, the difference between the two compressors is negligible, on others, the differential is substantial. Therefore, in BWIC_I, we try both JPEG 2000 and BWIC on each preprocessed channel and select and record the one with the best compression gain (see Fig. 5). This double processing, which is easily parallelizable, results in small overhead since only the hierarchically preprocessed quarter-sized image, not the entire chrominance channel, needs to be compressed. Moreover, it is possible to predict with high probability the most effective algorithm (JPEG 2000 versus BWIC) by examining a much smaller sized subimage of appropriately sampled pixels, as we will report in Sec. 4. Decompressing is done only once with the already selected compressor. Thus, there is no computational overhead in decompressing.

4 Experimental Results

This section presents quantitative results on the performance of the proposed lossless image compression technique BWIC_I versus those of the lossless compression schemes interband CALIC, M-CALIC, BPG (based on HEVC intra-frame encoder), JPEG-LS, JPEG 2000, BWIC, and LCIC using a collection of test images. In Ref. 17, it is demonstrated experimentally that LCIC currently achieves the best average compression ratios among these schemes. Therefore, for a fair comparison of the proposed scheme BWIC_I with LCIC, we have used the same set of test images Kodak, Medical, Commercial Digital Camera, and Classic used by Kim and Cho.[17]

For the convenience of the reader, Table 1 lists the references to the publicly available executable codes of most of these compressors and the parameters used. The compression results of these schemes on the test image sets are tabulated in Tables 2–5. The compression results of interband CALIC and M-CALIC on select images are indicated in the text. The performance of compressors on the test images is given in bits per pixel (bpp), which is defined as follows:

$$\text{bpp} = \frac{\text{Compressed file size} \times 8}{\text{Number of pixels}}. \tag{4}$$

Before delving into the details of performance tables, we present a brief experimental result on interband CALIC. We were informed by the authors of interband CALIC that the executable program is no longer available. Therefore, to measure the performance of BWIC_I against interband CALIC, we used the two ITU-standardized RGB images, cats and water,[45] used by Wu and Memon.[20] As listed in Table 1 of the aforementioned publication, while interband CALIC yielded 1.81 bpp on cats and 1.51 bpp on water, our algorithm achieved 1.53 bpp on cats and 1.31 bpp on water—a 17.2% improvement on average.

On the Kodak set in Table 2, the proposed technique surpassed BPG 63.6%, JPEG-LS 52.5%, BWIC 29.7%, JPEG 2000 10.7%, and LCIC 3.0%. Additionally, it was reported in Table 1 of Ref. 16 that M-CALIC yielded, on average, 10.4954 bpp on the Kodak image set; our proposed algorithm achieves 8.6042 bpp, which is 21.9% improvement over M-CALIC.

In Table 2, each image file was compressed in its entirety with the indicated scheme. It is also possible to preprocess RGB channels with an invertible RCT similar to one in Sec. 2 before employing a compressor. Our experimental results on the Kodak set showed mixed results when RCT preprocessing was used. While the compression of RGB channels improved in JPEG-LS 32.1% from 13.1239 to 9.9370 bpp, BWIC 16.4% from 11.1625 to 9.5890 bpp, and in JPEG 2000, the preprocessing of images has resulted 8.7% decrease in coding performance from 9.5254 to 10.4377 bpp. BPG results are not available since it does not support 16 bits/channel coding. In any case, the average 8.6042 bpp of BWIC_I remains the smallest.

The gains of the proposed technique were most substantial for the medical image set as seen in Table 3, where the proposed technique surpassed BPG, JPEG-LS, JPEG 2000, and LCIC more than 10%. We note that although BWIC is a universal compressor, as reported in Ref. 28, it yields good compression on medical images. Yet, except with the images with a low number of different colors, the proposed BWIC_I yields 6.7% gain over BWIC on average.

On the commercial digital camera image set in Table 4, the proposed technique bettered BPG, JPEG-LS, and BWIC more than 31%, while it surpassed JPEG 2000 by 8.6% and LCIC by 2.7%. For the classic image set in Table 5, while the

Table 1 Codec list and parameter settings.

Codec	Version	Parameters	Options
CALIC[38]	N/A	-in kodak01_Y.pgm -out kodak01_Y.calic	Use default options
			Binary mode enabled
BPG[39]	0.9.8	-lossless kodak01.png -o kodak01.bpg	Use default options
			q 0
			f 444
			c ycbcr
			b 8
			lossless
			e x265
			m 8
JPEG-LS[40]	JLS Encode 1.0	N/A (GUI) (kodak01.ppm)	Use default options
			Lossless compression
			Error value: 0
			Interleaved mode for color images
			Line interleaved mode advance settings
			Use default reset value: 64
			Use default threshold values
			T1 value: 3
			T2 value: 7
			T3 value: 21
JPEG 2000[41]	libjasper 1.900.1	–input kodak01.ppm –output kodak01.jp2	N/A
BWIC[42]	N/A	kodak01.ppm kodak01.bwic	N/A
LCIC[43]	N/A	-e kodak01.ppm kodak01.bin	N/A
Proposed BWIC_ I[44]	N/A	-e kodak01.ppm kodak01.bwici	N/A

proposed technique outperformed JPEG-LS by 2.6%, BPG by 11%, BWIC by 31%, and LCIC by 1.2%, it did no better than JPEG 2000.

The average encoding and decoding times of the proposed algorithm BWIC_I are benchmarked against JPEG 2000 and LCIC with an Intel Core Processor i7-940; the results for all the image sets used in this paper are tabulated in Table 6. It is evident that LCIC requires more time than JPEG 2000, as already stated in Ref. 17: "Since our method employs JPEG 2000 and needs additional steps for hierarchical prediction and context modeling, it needs slightly more time than the JPEG 2000." An inspection of Table 6 shows that the encoding and decoding times of our algorithm are slightly slower than LCIC, except for the decoding times of the commercial digital camera image set containing high-resolution pictures.

It is possible to shorten the encoding times of the proposed algorithm by using an iterative sampling technique. Our algorithm employs two different compression techniques, JPEG 2000 and BWIC, on the preprocessed chrominance channels and selects the best one. To determine the best such algorithm, one can construct a smaller image consisting of the pixels P_{ij} with i, j even of the original image, as shown in Fig. 8, and select the best compression techniques for the preprocessed chrominance channels of the sampled subimage, which is only a quarter of the original image. Our numerical experiments on the test image sets showed that the choice of the best compressor for the preprocessed chrominance channels of the sampled subimage agreed with 95.5% accuracy with the choices for the original larger image. When we iterated the same sampling process on the quarter-sized subimage, we observed 86.4%

Table 2 Results of various compression techniques in bpp for Kodak image set.[44]

	Size	BPG (HEVC)[39]	JPEG-LS (LOCO-I)[40]	JPEG 2000[41]	BWIC[42]	LCIC[43]	Proposed BWIC_ I[44]
Kodak01	768 × 512	17.0877	15.8493	10.3844	11.3110	9.5626	9.2833
Kodak02	768 × 512	13.0859	12.1827	9.1628	8.4586	8.4401	8.0753
Kodak03	768 × 512	11.0538	10.5267	8.0917	8.4873	7.3870	7.1228
Kodak04	768 × 512	13.4017	12.6077	9.1116	10.9012	8.3412	8.1542
Kodak05	768 × 512	16.8107	15.5738	10.8167	14.7568	10.2784	9.9278
Kodak06	768 × 512	15.0310	13.7333	9.5911	10.8477	8.9725	8.8295
Kodak07	768 × 512	11.7865	10.9859	8.5039	9.3731	7.7329	7.4525
Kodak08	768 × 512	17.2057	15.8642	11.1389	14.7589	10.6289	10.2662
Kodak09	768 × 512	12.5061	12.0051	8.9045	9.5114	7.9530	7.6783
Kodak10	768 × 512	12.5541	12.0296	9.0564	10.2927	8.1953	7.9259
Kodak11	768 × 512	14.2856	13.2320	9.2918	11.1251	8.5929	8.3059
Kodak12	768 × 512	12.3141	11.5165	8.6577	8.9579	7.8234	7.6286
Kodak13	768 × 512	19.3857	17.8546	11.8608	14.4810	11.3189	11.0565
Kodak14	768 × 512	16.1841	14.7745	10.1605	12.7430	9.5039	9.2534
Kodak15	768 × 512	12.2268	11.7120	8.9967	10.3740	8.3183	8.0809
Kodak16	768 × 512	13.4354	12.2641	8.7748	8.4977	7.9816	7.8081
Kodak17	768 × 512	13.1573	12.4618	9.0644	9.5929	8.1645	7.8597
Kodak18	768 × 512	16.4933	15.2824	10.7706	14.1720	10.3275	10.0852
Kodak19	768 × 512	14.4711	13.5989	9.6655	11.0480	8.9783	8.7657
Kodak20	768 × 512	9.6847	9.2185	8.0769	8.2231	7.8672	7.6284
Kodak21	768 × 512	14.6310	13.6095	9.7621	11.1269	9.1786	8.9921
Kodak22	768 × 512	14.9288	13.6449	10.0939	13.8337	9.7349	9.5373
Kodak23	768 × 512	11.4152	10.6474	8.5047	12.6672	7.8049	7.6823
Kodak24	768 × 512	14.7175	13.7982	10.1673	12.3583	9.5231	9.1009
Average		14.0772	13.1239	9.5254	11.1625	8.8587	8.6042
Normalized		1.636	1.525	1.107	1.297	1.030	1.000

agreement. This sampling process shortens the encoding times of our proposed algorithm by avoiding the need to try two compressors on the preprocessed chrominance channels of the much larger original image.

At the expense of negligible additional encoding times, it is possible to improve the compression gains of our proposed technique by enhancing the efficiency of BWIC. While JPEG 2000 is a two-dimensional image compressor, BWIC is a general-purpose compressor for one-dimensional data. Therefore, the choice of the scanning technique to convert a two-dimensional image to a one-dimensional sequence can have a substantial effect on the performance of BWIC, as investigated in Refs. 46 and 47.

We have experimented with four different scanning techniques, as depicted in Fig. 9, on the chrominance channels of the test image sets. On average, choosing the best scanning technique for BWIC resulted in an additional compression gain of 0.4% on Kodak and 0.6% on the medical image set. There is a large number of possible scanning paths[48,49] and the prediction of the optimal one without trying them all appears to be difficult. The compression gains can be further improved when coding schemes in Refs. 50–52 are utilized at the expense of a considerably significant encoding time. These compression algorithms require more computation time to better predict the pixel values compared to JPEG 2000, for example, as the authors state in Ref. 50: “when

Table 3 Results of various compression techniques in bpp for medical image set.[44]

	Size	BPG (HEVC)[39]	JPEG-LS (LOCO-I)[40]	JPEG 2000[41]	BWIC[42]	LCIC[43]	Proposed BWIC_ I[44]
PET1	256 × 256	4.8594	4.5022	6.7390	1.1975	5.6453	4.6204
PET2	256 × 256	5.5095	4.8088	7.3403	1.3191	6.1598	5.0226
PET3	256 × 256	5.1495	4.4931	7.0232	1.2716	5.8768	4.7799
Eye1	3216 × 2136	7.2370	5.9600	5.7498	5.9360	4.6208	4.4629
Eye2	3216 × 2136	6.4587	5.2964	5.4467	5.0913	4.3350	4.1568
Eyeground	1600 × 1216	5.7181	4.3317	3.2763	3.8550	2.9656	2.6873
Endoscope1	603 × 552	11.3375	8.0832	7.3532	12.1131	7.0917	6.9229
Endoscope2	568 × 506	7.1962	4.8612	5.1304	9.1321	4.8968	4.7718
Average		6.6832	5.2921	6.0074	4.9895	5.1990	4.6781
Normalized		1.429	1.131	1.284	1.067	1.111	1.000

Table 4 Results of various compression techniques in bpp for commercial digital camera image set.[44]

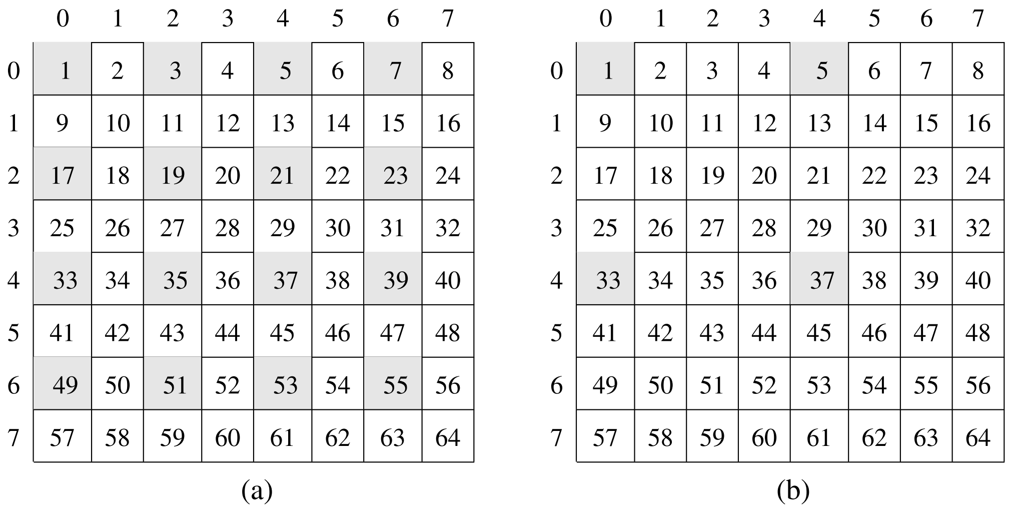
	Size	BPG (HEVC)[39]	JPEG-LS (LOCO-I)[40]	JPEG 2000[41]	BWIC[42]	LCIC[43]	Proposed BWIC_ I[44]
Berry	4288 × 2848	10.5564	9.0748	7.2468	11.0263	6.8917	6.6871
Ceiling	4288 × 2848	11.6345	10.2694	7.5571	11.9231	7.2080	7.0403
Fireworks	4032 × 3024	8.0156	6.5926	5.7797	7.3418	5.2855	5.1709
Flamingo	4288 × 2848	8.6922	7.9900	7.0366	8.3427	6.6371	6.4789
Flower	4032 × 3024	10.0319	7.9303	6.4141	15.6980	6.0655	5.9508
Locks	4288 × 2848	11.0121	9.8343	7.4574	10.5541	7.1623	6.9365
Park	4032 × 3024	9.0735	7.8993	5.8977	7.4954	5.5622	5.4509
Sunset	4288 × 2848	8.4887	7.3256	6.3586	6.9975	5.9700	5.7550
Average		9.6881	8.3645	6.7185	9.9224	6.3478	6.1838
Normalized		1.567	1.353	1.086	1.605	1.027	1.000

Table 5 Results of various compression techniques in bpp for classic image set.[44]

	Size	BPG (HEVC)[39]	JPEG-LS (LOCO-I)[40]	JPEG 2000[41]	BWIC[42]	LCIC[43]	Proposed BWIC_ I[44]
Lena	512 × 512	14.4364	13.5519	13.5848	17.5698	13.6461	13.4594
Peppers	512 × 512	15.0658	14.2113	14.8000	19.0659	15.2102	15.0028
Mandrill	512 × 512	20.2404	18.5386	18.0939	21.5096	18.5305	18.3218
Barbara	640 × 512	14.7933	13.3149	11.1612	18.0198	11.4575	11.3425
Average		16.1340	14.9042	14.4100	19.0413	14.7111	14.5316
Normalized		1.110	1.026	0.992	1.310	1.012	1.000

Table 6 Comparison of average processing times (s) of image sets.

	Encoding			Decoding		
	JPEG 2000	LCIC	Proposed	JPEG 2000	LCIC	Proposed
Kodak	0.6092	0.7832	1.0292	0.5124	0.7086	0.9026
Medical	2.1261	2.8919	3.4014	1.8023	2.7430	2.8871
Commercial	13.7639	18.2223	19.4103	11.5839	17.0603	15.3935
Classic	0.5575	0.6128	0.8743	0.4665	0.5580	0.7653



(a)

	0	1	2	3	4	5	6	7
0	**1**	2	**3**	4	**5**	6	**7**	8
1	9	10	11	12	13	14	15	16
2	**17**	18	**19**	20	**21**	22	**23**	24
3	25	26	27	28	29	30	31	32
4	**33**	34	**35**	36	**37**	38	**39**	40
5	41	42	43	44	45	46	47	48
6	**49**	50	**51**	52	**53**	54	**55**	56
7	57	58	59	60	61	62	63	64

(b)

	0	1	2	3	4	5	6	7
0	**1**	2	3	4	**5**	6	7	8
1	9	10	11	12	13	14	15	16
2	17	18	19	20	21	22	23	24
3	25	26	27	28	29	30	31	32
4	**33**	34	35	36	**37**	38	39	40
5	41	42	43	44	45	46	47	48
6	49	50	51	52	53	54	55	56
7	57	58	59	60	61	62	63	64

Fig. 8 (a) and (b) Highlighted pixels used in the iterative sampling technique for an image with 64 pixels.

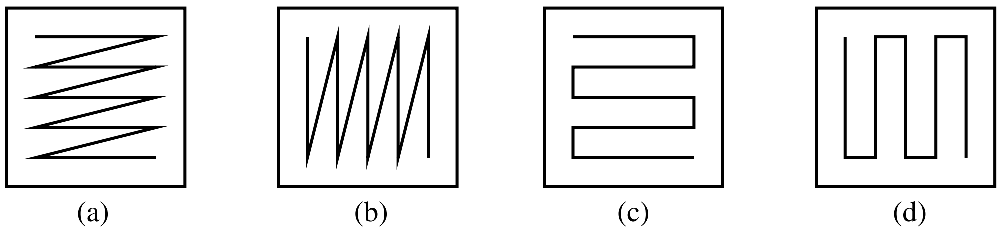


Fig. 9 (a)–(d) Raster and snake-like scanning paths.

the size of images is 512 × 512 pixels, for example, our codec takes 10 to 25 min for encoding and at most 0.2 s for decoding on a computer with the 3.06 GHz Xeon processor." In our proposed compression algorithm, the encoding time is on average less than 5 s for an image file.

5 Conclusions

In this study, we proposed a coding technique BWIC_I for lossless compression of color images that extends the hierarchical prediction technique LCIC of Kim and Cho.[17] We demonstrated that, after transforming an image in RGB domain into (YC_uC_v) channels, the use of CALIC on Y, and a BWIC or JPEG 2000, along with a hierarchical prediction technique on C_u and C_v improves compression gain. We reported the compression ratios, encoding and decoding runtimes of the proposed technique for a multitude of image sets. Compression gains over JPEG 2000 were 10.7% on Kodak, 28.4% on medical, and 8.6% on commercial digital camera image sets. These compression gains with the proposed technique were on average a 4.5% improvement over the gains reported in Ref. 17; on the medical image set, in particular, this improvement increased to 11.1%. In addition, we have shown that the compression gain in the proposed compression scheme can be further improved by another 0.4% to 0.6% by using a suitable scanning technique.

References

1. B. Koc et al., "A new technique for lossless compression of color images based on hierarchical prediction, inversion and context adaptive coding," in *Data Compression Conf. (DCC)*, pp. 584–584 (2019).
2. M. U. Celik, G. Sharma, and A. M. Tekalp, "Gray-level-embedded lossless image compression," *Signal Process. Image Commun.* **18**(6), 443–454 (2003).

3. S. Dikbas and F. Zhai, "Lossless image compression using adjustable fractional line-buffer," *Signal Process. Image Commun.* **25**(5), 345–351 (2010).
4. V. Anusuya, V. S. Raghavan, and G. Kavitha, "Lossless compression on MRI images using SWT," *J. Digital Imaging* **27**, 594–600 (2014).
5. K. C. Pathak and J. N. Sarvaiya, "Lossless medical image compression using transform domain adaptive prediction for telemedicine," in *Int. Conf. Wireless Commun., Signal Process. and Networking (WiSPNET)*, pp. 1026–1031 (2017).
6. S. K. Mohammed, K. M. Rahman, and K. A. Wahid, "Lossless compression in Bayer color filter array for capsule endoscopy," *IEEE Access* **5**, 13823–13834 (2017).
7. M. Hernandez-Cabronero et al., "Lossless compression of color filter array mosaic images with visualization via JPEG 2000," *IEEE Trans. Multimedia* **20**(2), 257–270 (2018).
8. F. Liu et al., "The current role of image compression standards in medical imaging," *Information* **8**(4), 131 (2017).
9. M. D. Adams and F. Kossentni, "Reversible integer-to-integer wavelet transforms for image compression: performance evaluation and analysis," *IEEE Trans. Image Process.* **9**(6), 1010–1024 (2000).
10. H. S. Malvar, G. J. Sullivan, and S. Srinivasan, "Lifting-based reversible color transformations for image compression," *Proc. SPIE* **7073**, 707307 (2008).
11. S. Pei and J. Ding, "Improved reversible integer-to-integer color transforms," in *16th IEEE Int. Conf. Image Processing (ICIP)*, pp. 473–476 (2009).
12. P. Hao and Q. Shi, "Matrix factorizations for reversible integer mapping," *IEEE Trans. Signal Process.* **49**(10), 2314–2324 (2001).
13. T. Strutz, "Multiplierless reversible color transforms and their automatic selection for image data compression," *IEEE Trans. Circuits Syst. Video Technol.* **23**(7), 1249–1259 (2013).
14. S. Zhu et al., "Cross–space distortion directed color image compression," *IEEE Trans. Multimedia* **20**(3), 525–538 (2018).
15. Information Technology, "JPEG 2000 image coding system-part 1: core coding system," INCITS/ISO/IEC Standard 15444–1 (2000).
16. S. Kim and N. I. Cho, "A lossless color image compression method based on a new reversible color transform," in *Visual Commun. and Image Process.*, pp. 1–4 (2012).
17. S. Kim and N. I. Cho, "Hierarchical prediction and context adaptive coding for lossless color image compression," *IEEE Trans. Image Process.* **23**, 445–449 (2014).
18. ITU-T and ISO/IEC, "JPEG XR image coding system-Part 2: image coding specification," ISO/IEC Standard 29199–2 (2011).
19. X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," *IEEE Trans. Commun.* **45**(4), 437–444 (1997).
20. X. Wu and N. Memon, "Context-based lossless interband compression-extending CALIC," *IEEE Trans. Image Process.* **9**(6), 994–1001 (2000).
21. E. Magli, G. Olmo, and E. Quacchio, "Optimized onboard lossless and near-lossless compression of hyperspectral data using CALIC," *IEEE Geosci. Remote Sens. Lett.* **1**(1), 21–25 (2004).
22. M. Rabbani and R. Joshi, "An overview of the JPEG 2000 still image compression standard," *Signal Process. Image Commun.* **17**(1), 3–48 (2002).
23. A. Khademi and S. Krishnan, "Comparison of JPEG 2000 and other lossless compression schemes for digital mammograms," in *IEEE Eng. Med. and Biol. 27th Annu. Conf.*, pp. 3771–3774 (2005).
24. D. Taubman and M. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*, Springer, Boston, Massachusetts (2013).
25. M. Burrows and D. Wheeler, "A block-sorting lossless data compression algorithm," Technical Report, Digital SRC Research Report (1994).
26. E. Syahrul et al., "Lossless image compression using Burrows Wheeler Transform (methods and techniques)," in *IEEE Int. Conf. Signal Image Technol. and Internet Based Syst.*, pp. 338–343 (2008).
27. M. Schindler, "A fast block-sorting algorithm for lossless data compression," in *Proc. Data Compression Conf.*, p. 469 (1997).
28. Z. Arnavut, "Inversion coding," *Comput. J.* **47**(1), 46–57 (2004).
29. J. L. Bentley et al., "A locally adaptive data compression scheme," *Commun. ACM* **29**(4), 320–330 (1986).
30. P. Elias, "Interval and recency rank source coding: two on-line adaptive variable-length schemes," *IEEE Trans. Inf. Theory* **33**(1), 3–10 (1987).
31. Z. Arnavut and M. Arnavut, "Investigation of block-sorting of multiset permutations," *Int. J. Comput. Math.* **81**(10), 1213–1222 (2004).
32. P. M. Fenwick, "The Burrows–Wheeler transform for block sorting text compression: principles and improvements," *Comput. J.* **39**(9), 731–740 (1996).
33. B. Balkenhol and S. Kurtz, "Universal data compression based on the Burrows–Wheeler transformation: theory and practice," *IEEE Trans. Comput.* **49**(10), 1043–1053 (2000).
34. P. Fenwick, "Burrows–Wheeler compression: principles and reflections," *Theor. Comput. Sci.* **387**(3), 200–219 (2007).
35. S. Deorowicz, "Improvements to Burrows–Wheeler compression algorithm," *Software Pract. Exper.* **30**(13), 1465–1483 (2000).
36. M. J. Weinberger, G. Seroussi, and G. Sapiro, "LOCO-I: a low complexity, context-based, lossless image compression algorithm," in *Data Compression Conf. Proc.*, IEEE, pp. 140–149 (1996).
37. R. Starosolski, "Skipping selected steps of DWT computation in lossless JPEG 2000 for improved bitrates," *PloS One* **11**, e0168704 (2016).
38. Game Closure, "The implementation of CALIC algorithm," 2013, https://github.com/gameclosure/gcif/tree/master/refs/calic (accessed 16 July 2019).
39. F. Bellard, "The implementation of BPG image format," 21 April 2018, https://bellard.org/bpg (accessed 16 July 2019).
40. I. R. Ismail and F. Kossentini, "UBC's JPEG-LS Codec implementation v.2.2," 1997, http://www.stat.columbia.edu/~jakulin/jpeg-ls/mirror.htm (accessed 16 July 2019).
41. M. Adams, "JasPer implementation of JPEG 2000, version 1.900.1," 2006, https://www.ece.uvic.ca/~frodo/jasper/ (accessed 16 July 2019).
42. Z. Arnavut, "The implementation of Burrows-Wheeler with Inversion Coder (BWIC)," 2004, http://www.cs.fredonia.edu/arnavut/research.html (accessed 16 July 2019).
43. S. Kim and N. I. Cho, "The implementation of LCIC," January 2014 http://ispl.snu.ac.kr/light4u/project/LCIC/ (accessed 16 July 2019).
44. B. Koc et al., "The implementation of BWIC I," 2019, http://www.cs.fredonia.edu/arnavut/research/BWIC_I (accessed 16 July 2019).
45. ITU-T Test Signals for Telecommunication Systems, "Rec. ITU-T T.24 Standardized digitized image set," November 2015, https://www.itu.int/net/itu-t/sigdb/genimage/test24.htm (accessed 16 July 2019).
46. Z. Arnavut, B. Koc, and H. Kocak, "Scanning paths for lossless compression of pseudo-color images," in *IEEE Western New York Image and Signal Process. Workshop (WNYISPW)*, pp. 11–14 (2014).
47. C. Preston, Z. Arnavut, and B. Koc, "Lossless compression of medical images using Burrows-wheeler transformation with inversion coder," in *37th Annu. Int. Conf. IEEE Eng. Med. and Biol. Soc. (EMBC)*, pp. 2956–2959 (2015).
48. N. D. Memon, K. Sayood, and S. S. Magliveras, "Lossless image compression with a codebook of block scans," *IEEE J. Sel. Areas Commun.* **13**(1), 24–30 (1995).
49. H. Liu et al., "Quadtree coding with adaptive scanning order for space-borne image compression," *Signal Process. Image Commun.* **55**, 1–9 (2017).
50. I. Matsuda et al., "Lossless coding using variable block-size adaptive prediction optimized for each image," in *13th Eur. Signal Process. Conf.*, pp. 1–4 (2005).
51. T. Strutz, "Context-based predictor blending for lossless color image compression," *IEEE Trans. Circuits Syst. Video Technol.* **26**(4), 687–695 (2016).
52. Y. Zhang and D. A. Adjeroh, "Prediction by partial approximate matching for lossless image compression," *IEEE Trans. Image Process.* **17**(6), 924–935 (2008).

Basar Koc received his BSc degree in computer science from State University of New York at Fredonia and received his MSc and PhD degrees in computer science from the University of Miami. He is an assistant professor at Stetson University. He is primarily interested in data and image compression.

Ziya Arnavut received his BSc degree in applied mathematics from Ege University in 1983 and received his MSc degree in computer science from the University of Miami and PhD in computer science from the University of Nebraska-Lincoln in 1987 and 1995, respectively. Since August 1997, he has worked as a professor in the Computer and Information Sciences Department of SUNY Fredonia. His research interests are data compression, algorithms, and image processing.

Dilip Sarkar received his BTech degree from the Indian Institute of Technology, Kharagpur, the MSc degree from the Indian Institute of Science, Bangalore, and the PhD degree from the University of Central Florida, Orlando. He is currently an associate professor in the Department of Computer Science, University of Miami, Coral Gables, Florida. His research interests include VBR video traffic modeling, multimedia communication over broadband and wireless networks, middleware and web computing, design and analysis of algorithms, and parallel and distributed processing.

Hüseyin Koçak received his doctoral degree from the University of California. He is now a professor of computer science and mathematics at the University of Miami. His research interests include scientific computing, computer graphics, and theory and applications of dynamical systems to biology, medicine, marine, and atmospheric sciences.