Multimedia Tools and Applications (2022) 81:4781–4795
https://doi.org/10.1007/s11042-021-11017-5

1193: INTELLIGENT PROCESSING OF MULTIMEDIA SIGNALS

An improved lossless image compression algorithm based on Huffman coding

Xiaoxiao Liu[1] · Ping An[1] · Yilei Chen[1] · Xinpeng Huang[1]

Received: 15 October 2020 / Revised: 25 February 2021 / Accepted: 5 May 2021 /
Published online: 8 June 2021

Abstract
There is an increasing number of image data produced in our life nowadays, which creates a big challenge to store and transmit them. For some fields requiring high fidelity, the lossless image compression becomes significant, because it can reduce the size of image data without quality loss. To solve the difficulty in improving the lossless image compression ratio, we propose an improved lossless image compression algorithm that theoretically provides an approximately quadruple compression combining the linear prediction, integer wavelet transform (IWT) with output coefficients processing and Huffman coding. A new hybrid transform exploiting a new prediction template and a coefficient processing of IWT is the main contribution of this algorithm. The experimental results on three different image sets show that the proposed algorithm outperforms state-of-the-art algorithms. The compression ratios are improved by at least 6.22% up to 72.36%. Our algorithm is more suitable to compress images with complex texture and higher resolution at an acceptable compression speed.

Keywords Lossless compression · Linear prediction · Integer wavelet transform · Huffman coding

1 Introduction

It is a crucial challenge to develop image compression techniques because of the increasing high-resolution and high-quality images produced in the present day, but the bandwidth and storage capacity are limited. Image compression is widely used in many fields such as broadcast television, aircraft, teleconferencing, and computer transmission. Image compression is divided into two types: lossy and lossless [14]. In lossy compression, as the name states that some details of the original image will be lost after compression. On the contrary, lossless

✉ Ping An
anping@shu.edu.cn

[1] Shanghai Institute for Advanced Communication and Data Science, School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China

compression remains the same as the original image after decompression. Lossless image compression is most frequently used in some fields requiring high fidelity, such as medical imaging, military communication via radar, and remote sensing applications. Taking medical imaging as an example, modern medical diagnosis relies heavily on digital images in many modalities such as computed tomography (CT), X-ray, magnetic resonance imaging (MRI), and ultrasound (US). To analysis human health problems accurately, these medical images should be saved in a lossless format without any loss of crucial parts. In the case of remote medical diagnosis, an excellent lossless compression technique can save transmission time to improve treatment efficiency.

For lossless image compression, a lot of research work has been carried out [25]. The state-of-the-art lossless image compression algorithms are Run-Length Encoding (RLE) [12], the entropy coding [9], and the dictionary-based coding [17]. In RLE, data is stored as a single value and the count of the same consecutive values, so the code length is decreased significantly. Run-length coding works well for data with long runs of identical samples. However, RLE fails when it comes to authentic images because of inconsecutive pixel values. The entropy coding, such as Shannon-Fano, Huffman and arithmetic coding, achieves better results of image compression than RLE. Shannon-Fano coding takes the sorted probabilities of image data in descending order, then divides them into two groups while keeping the sum of each group almost equal. The former steps are repeated until each element becomes a leaf node on a binary tree. Sometimes Shannon-Fano coding cannot reproduce data perfectly and be poor in compression ratio because it cannot produce an optimal binary tree. Huffman coding makes up this shortcoming. Huffman coding forms the binary tree of the sorted probabilities from the leaves to the root, which is opposite to Shannon-Fano [31]. Huffman coding can provide an optimal code [26], but it is very sensitive to noise. Additionally, arithmetic coding presents data into a fixed decimal range based on a given probability set, then only saves the upper and lower limit of the range. Arithmetic coding provides a better compression performance [7], but the compression speed is low. Furthermore, the original image can be corrupted if a single bit error occurs in arithmetic coding. Lastly, the dictionary-based coding methods use a look-up table to encode a sequence of characters. Lempel-Ziv-Welch (LZW) is a universal lossless compression algorithm created by Abraham Lempel, Jacob Ziv, and Terry Welch. It is an updated version of LZ77 and LZ78. LZW algorithm builds a dictionary to map the multiple occurrences of repetitive substrings [18]. It is suitable for deducing files size that carry more repeated content, but it is more complicated and time-consuming if data is large and discontinuous [39].

Nowadays, some frequently used algorithms are proposed by combining the above coding methods, such as Deflate, CALIC, JPEG-LS, and JPEG 2000. Deflate is a combination of the LZ77 algorithm and Huffman coding. The results in [23] claim that Deflate algorithm is more efficient than LZW in both compression ratio and speed. Context-based, adaptive, lossless image codec (CALIC) is a sequential coding scheme, which uses modeling context with arithmetic coding. CALIC is a very efficient lossless compression algorithm with a high compression ratio [33, 38]. JPEG-LS is the ISO/ITU standard for lossless and near-lossless compression of continuous-tone images. It is based on a variation of the low complexity lossless compression method (LOCO-I) [29], and use context models in conjunction with Golomb coding. JPEG-LS yields better compression ratios compared to other relevant schemes reported in [37]. The standard JPEG 2000 is mostly based on some image transforms and EBCOT (Embedded Block Coding with Optimized Truncation of the embedded bitstreams), offering good compression performance than others [6], and the compressed version of the image in JPEG 2000 shows better in detail than JPEG [10].

Palette reordering is one of the main approaches for improving the lossless compression of color-indexed images. Color-indexed images are defined by a matrix of indexes and a color palette (color-map). Each index points to a position in the color-map. Palette reordering aims to improve the lossless compression rates through finding an optimal permutation of the color palette [24]. In [11], Oliver Giudice et al. propose a novel re-indexing approach where the traveling salesman problem is solved with ant colony optimization (TSP_ACO). The method outperforms state-of-the-art ones in terms of compression gain. To further improve the TSP_ACO method, Shahrukh Sheikh and Athi Narayanan [32] apply a circular shift to the optimal color palette for M colors after the application of ant colony optimization, which improves the compression gain of the previous state-of-the-art methods.

Deep learning models have been recently explored for lossless image compression. A novel CNN-based prediction paradigm in [30] is specifically designed to predict highly textural regions in high-resolution images. Wu et.al presents a CNN-based predictive lossless compression scheme for raw color mosaic images of digital cameras in [4], which achieves unprecedented lossless compression performance on camera raw images. However, these learning-based methods use deep neural networks, including PixelCNN [2], PixelCNN++ [28], and Multiscale-PixelCNN [27], which always require hours on GPUs with high computation complexity and still need entropy encoding in the last step [22].

A new lossless image compression algorithm using a combination of two-dimensional discrete wavelet transform (DWT) and one-dimensional discrete fractional Fourier transform (DFrFT) is introduced by Kumar et al. [21]. They split an image into low-frequency sub-bands and high-frequency sub-bands. These two sub-bands are separately compressed by different methods. The results show that the method has a higher compression ratio and better image reconstruction quality.

A hybrid predictive lossless image compression algorithm is proposed by Azman et al. [5]. They find that the hybrid predictive technique in the sequence of DPCM-IWT-Huffman performs better. However, because the predictive formula and Harr wavelet used in the method are limited and simple, some spatial and spectral redundancy still have the probability to be further reduced. The limitations prevent the algorithm from giving the best entropy result and compression ratio.

The most important point of lossless compression method is to improve the compression ratio. However, many of the existing compression methods are complex in structure, and the improvement of compression ratio is still limited. In this paper, a novel lossless coding algorithm is proposed, which has a better compression performance in compression ratio with a simple structure. A new hybrid transform exploiting a new prediction template and a coefficient processing of IWT is the main contribution of this algorithm. The experimental results show that the proposed algorithm improve the efficiency of image lossless compression coding, and reduce the complexity of the compression algorithm compared with state-of-the-art lossless compression methods.

The novel contributions of this paper are summarized as follows:

- A new transform scheme is proposed to enhance the compression performance.
- The novel prediction template using 5 adjacent pixel values remove most of the interpixel redundancy and make for better compression ratio.
- Using LeGall 5/3 wavelet transforms to the prediction errors eliminates more redundancy of data.
- The additional procedure disposes of zero and negative values of wavelet coefficients.

- Our lossless algorithm is proved practical and effective in different image sets compared to many state-of-the-art methods.

The rest of this paper is organized as follows. Section 2 presents the details of our lossless compression algorithm. In section 3, the compression results of the proposed method compared with state-of-the-art algorithms are reported, proving the validity of our method. The analysis of the results is also included. Finally, the conclusions and future work are discussed in section 4.

2 Proposed method

In this section, we give details of our lossless image compression method, which combines linear predictive coding, integer wavelet transforms with coefficients processing, and Huffman coding. The flowchart of the proposed method encoder is shown in Fig. 1. First, the pixels of the input image are subjected to a 2-D linear prediction to obtain the integer pixel errors. Due to the great correlation between adjacent pixels, the error values obtained by the predictor are greatly reduced and even close to zero. This is conducive to the following wavelet transform to produce better coefficient distributions. In the next step, we apply the LeGall5/3 integer wavelet transform (IWT) to the prediction errors. We observed that the output coefficients of the IWT are very sparse, the number of 0 and the number of values with the same absolute value are very large. Therefore, the output coefficients of the IWT are processed to discard redundant zero and negative values, and record the positions in the data. The step of recording positions is quite important, because reconstructing the data exactly needs the accurate position information. After this step, the image data has been well processed that the entropy of it is greatly decreased, increasing the efficiency of the following entropy coding. Finally, the data is entropy encoded by Huffman coding to get the compressed bit stream.

2.1 2-D linear lossless prediction

Prediction is a crucial part of compression because it can remove most of the spatial redundancy between pixels, and the choice of an optimal predictor is essential for the efficiency of compression methods. Linear predictors have a significant advantage which is the possibility of realization of the integer system [3]. In our method, a proposed two-dimensional linear prediction method is used to remove the inter-pixel redundancy of images. To achieve the lossless transform, the predicted value will be rounded to the nearest integer so that the error value (the difference between the original and the predicted value) is also an integer. This ensures that the reconstructed image is identical to the original. The block diagram of a lossless linear predictive encoding and decoding system is shown in Fig. 2.

As seen from Fig. 2, the input image is firstly passed through the predictor where the adjacent pixels are calculated according to the set prediction template. Then the obtained predicted values are rounded to the nearest integer $\widehat{f}_n$. Finally, the integer predicted value $\widehat{f}_n$ is subtracted from the original pixel value f_n to obtain the integer error value e_n. Decoding is the inverse process of encoding. The prediction template of our method is shown in Fig. 3.

In Fig. 3, $I_{(i,j)}$ represents the pixel value at the position (i, j). In this step, the gray values of the top-left and bottom-right pixel of the input image are retained, and the gray values of other positions are calculated to obtain the predicted value $P_{(i,j)}$ as seen in eq. (1). Then the predicted

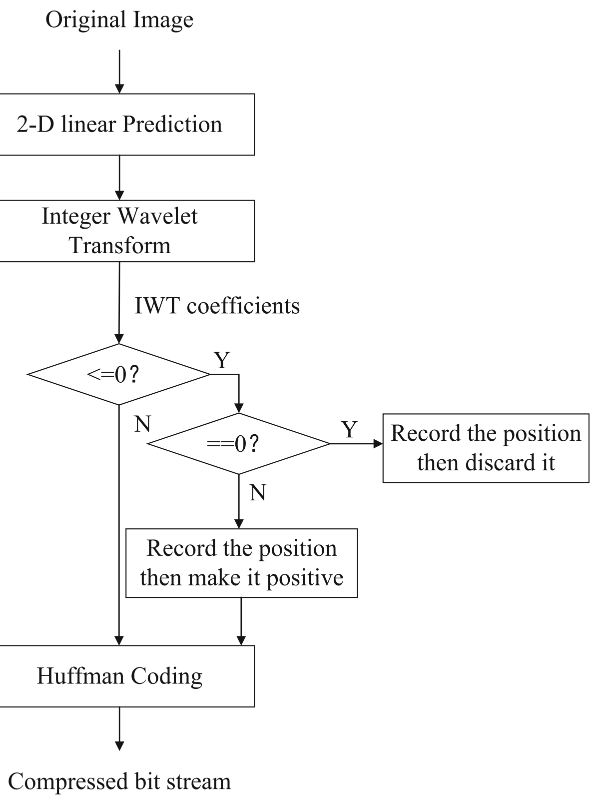

Fig. 1 Flowchart of the proposed method encoder

values are rounded in eq. (2). The error value $E_{(i,j)}$ is obtained in eq. (3) by subtracting the original pixel $I_{(i,j)}$ with the rounded predicted value $\overline{P}_{(i,j)}$, resulting in no loss of accuracy.

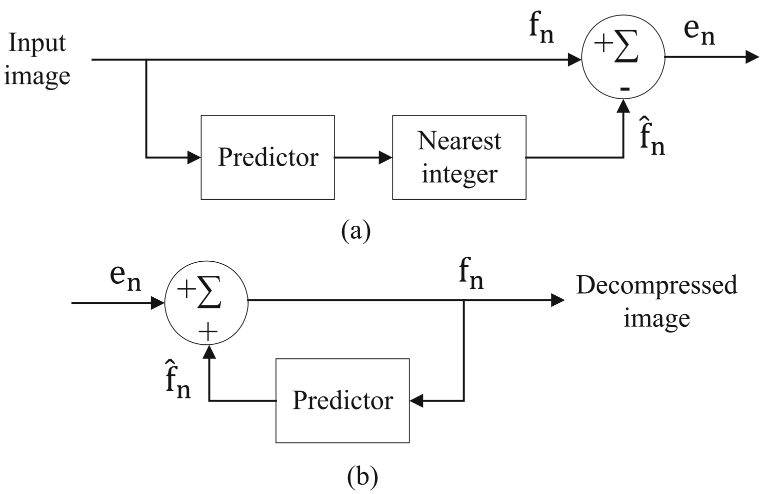

Fig. 2 Lossless predictive coding model: **a** Encoder; **b** Decoder

	$I_{(i-1,\,j-1)}$	$I_{(i-1,\,j)}$	$I_{(i-1,\,j+1)}$	
	$I_{(i,\,j-1)}$	$I_{(i,\,j)}$	$I_{(i,\,j+1)}$	

Fig. 3 Prediction template of the proposed method

$$P_{(i,j)} = \frac{1}{5}\left(I_{(i-1,j-1)} + I_{(i-1,j)} + I_{(i-1,j+1)} + I_{(i,j-1)} + I_{(i,j+1)}\right) \tag{1}$$

$$\overline{P}_{(i,j)} = \mathrm{round}\left(P_{(i,j)}\right) \tag{2}$$

$$E_{(i,j)} = I_{(i,j)} - \overline{P}_{(i,j)} \tag{3}$$

2.2 Integer wavelet transform (IWT)

The lifting scheme presented by Wim Sweldens [35] can yield reversible integer-to-integer wavelet transform. The basic lifting scheme consists of three stages: split, predict, and update (Fig. 4). Firstly, the input signal is decomposed into two disjoint sets with even and odd samples, respectively [16, 34]. In the predict phase, the predictor uses even samples $e_{j-1,k} = s_{j,2k}$ to predict odd samples $o_{j-1,k} = s_{j,2k+1}$. The prediction result d_{j-1} represents the high-frequency coefficients. In the update phase, the even samples e_{j-1} are updated with d_{j-1} to generate low-frequency coefficients s_{j-1}. Predict and update processes are performed in reverse order during the inverse transform. Finally, two disjoint sets are merged into one set to obtain ideal reconstruction data [19].

To select a suitable wavelet for lossless compression, we choose the 5/3 wavelet of Le Gall, because it can achieve integer-to-integer transform. The LeGall5/3 wavelet is a tightly supported biorthogonal wavelet with 2 vanishing moments [13], which can effectively eliminate the correlation between image pixels in a short time. The 5/3 filter is symmetrical, so it can avoid phase deformation during image processing. These features can ensure a satisfactory reconstruction of the original image.

The forward transform formulas of the 5/3 filter are shown in eqs. (4) and (5).

$$d_{j-1,k} = s_{j,2k+1} - \left\lfloor \frac{s_{j,2k} + s_{j,2k+2}}{2} \right\rfloor \tag{4}$$

$$s_{j-1,k} = s_{j,2k} + \left\lfloor \frac{d_{j-1,k-1} + d_{j-1,k+1} + 2}{4} \right\rfloor \tag{5}$$

where s_j is the error value $\overline{P}$ obtained in the predictive coding step, d_{j-1} and s_{j-1} are the high-frequency sub-band and low-frequency sub-band after the wavelet transform, respectively. To

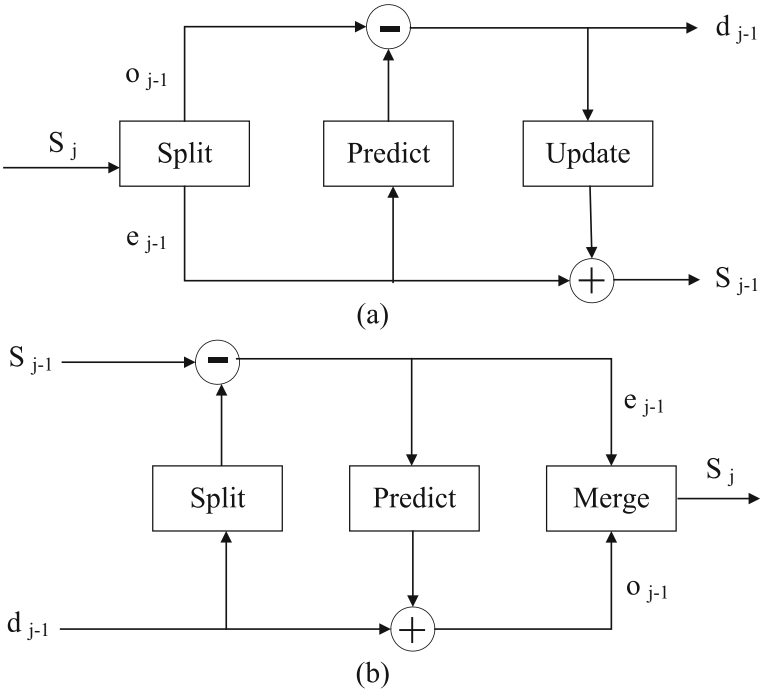

Fig. 4 The lifting scheme: **a** Forward transform; **b** Inverse transform

achieve the expected compression performance, the wavelet decomposition level selected in this paper is six. After 6 times' decompositions, the original data s_j is expressed as $\{s_{j-6}, \ d_{j-6}, \ d_{j-5}, \ \ldots, d_{j-1}\}$.

The inverse transform formulas of the LeGall5/3 integer wavelet are shown in eqs. (6) and (7).

$$s_{j,2k} = s_{j-1,k} - \left\lfloor \frac{d_{j-1,k-1} + d_{j-1,k+1} + 2}{4} \right\rfloor \tag{6}$$

$$s_{j,2k+1} = d_{j-1,k} + \left\lfloor \frac{s_{j,2k} + s_{j,2k+2}}{2} \right\rfloor \tag{7}$$

The inverse transform formulas are used in the decompression process, where $s_{j,\,2k}$ and $s_{j,\,2k+1}$ are the even sub-sequence and odd sub-sequence of the original data s_j. And they are combined to get the original data. The LeGall5/3 integer wavelet transform has only addition and subtraction during the transform [34]. Both forward and inverse transform have a precision loss during the rounding process. However, when there is more or less in the forward transform, correspondingly, the same amount will be reduced or increased in the inverse transform. This means there will be no loss of data. After rounding, the error before and after the transform is 0, so the purpose of lossless compression is achieved.

2.3 IWT coefficients processing

After images being processed by linear prediction and IWT, we execute the following simulations. For the exemplary image shown in Fig. 5(a), we compute the histogram of it in Fig. 5(b) and show the linear prediction output of the image and its histogram in Fig. 5(c) and Fig. 5(d), respectively. In Fig. 5(e) and Fig. 5(f), we show the IWT of linear prediction output and its histogram, respectively. We can see that the number of zeros increases with a strong symmetry in Fig. 5(f) more than others. Therefore, the proposed algorithm processes the transformed wavelet coefficients as follows.

- If the wavelet coefficient is positive, keep it;
- If the wavelet coefficient is 0, record the position information of 0 in the array x(i), then delete it;
- If the wavelet coefficient is negative, record the position information of the negative value in the array y(i), and then take the absolute value of the negative coefficient.

This method can increase the probability of data, which is conducive to the entropy coding to further reduce data redundancy and increase the compression ratio.

2.4 Huffman coding

To improve the compression ratio, entropy encoding is used to compress the above-transformed data further. Huffman coding yields better results in low algorithm complexity and good compression effect compared to others [20]. Therefore, Huffman encoding is adopted as entropy coding in our method. The basic procedure of Huffman coding is as follows:

- Calculate the probabilities of the source symbols in the image, and arrange them in descending order;
- Produce a node set by making these probabilities as the leaves of a binary tree;
- Take two nodes with the two lowest probabilities from the set, and generate a new probability representing the sum of these two probabilities;
- The new probabilities are reorganized in a descending order. Then repeat the last step until the node set contains only one node;
- A binary tree is created and Huffman codes are obtained from the tree assigning 0 and 1 to each left and right branch of the tree.

3 Experimental results and analysis

3.1 Datasets

We applied our algorithm to three different image sets to verify the effectiveness. The first set of images presented in Table 1 is from the waterloo image compression benchmark [15] having various kinds of graphics and images, which is shown in Fig. 6. Specifically, they consist of landscapes (“bridge” and “goldhill”), computer graphics (“circles”, “crosses”,

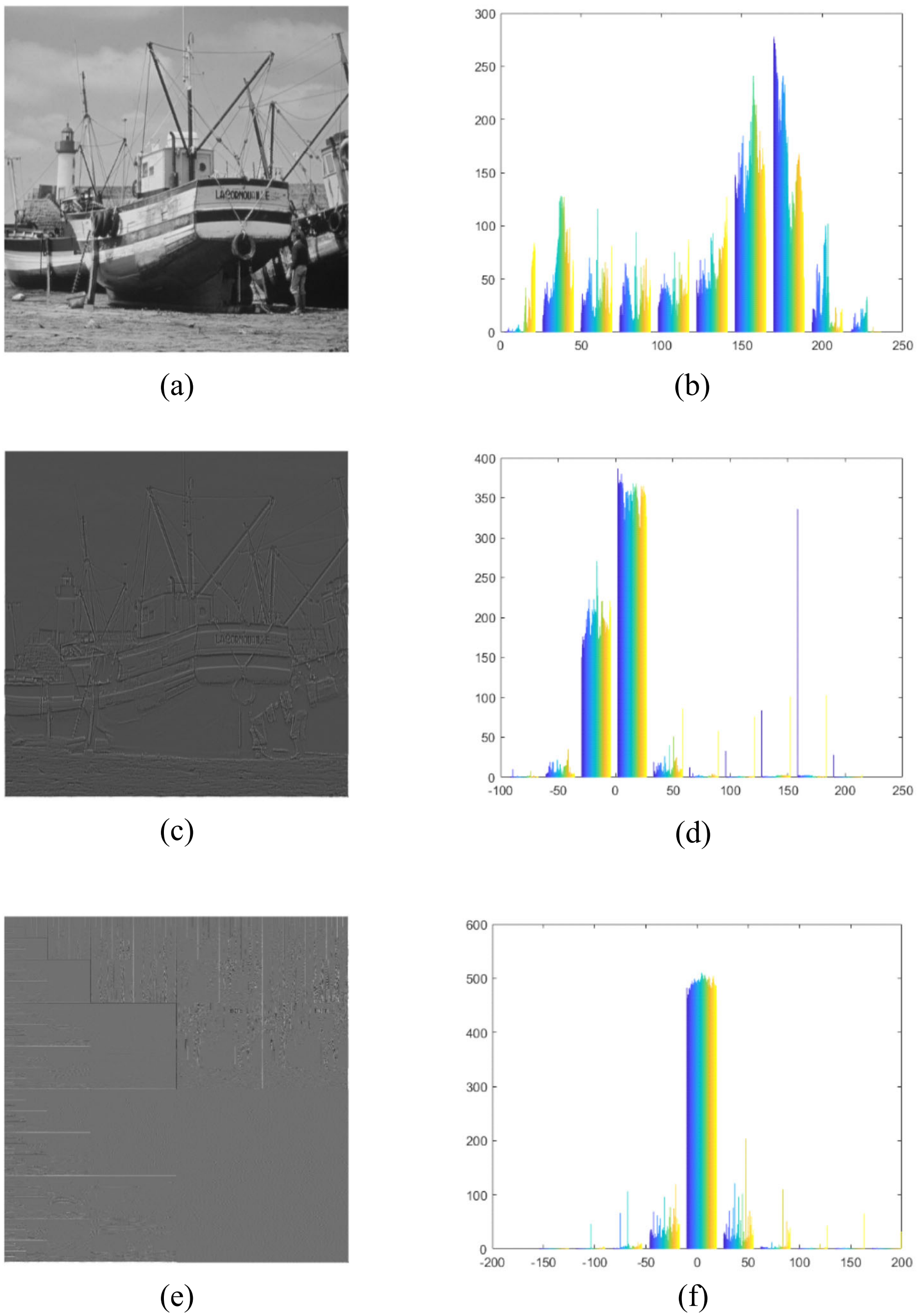

Fig. 5 **a** Original image; **b** Original image histogram; **c** Linear prediction output of the original image; **d** Linear prediction output of the original image histogram; **e** IWT of the linear prediction output; **f** IWT of the linear prediction output histogram

"slope", and "squares") and classic images of image processing literature ("lena", "mandrill" and "peppers"). Their sizes are from 256 × 256 pixels up to 512 × 512. The second set shown in Table 2 is the new test images with high-resolution and high-precision from Image Compression Benchmark [36]. These photographic images come from a wide variety of sources ranging from 2286 × 1512 pixels up to 7216 × 5412. In order to compare the compression results with Paq8px167 + CM proposed in [8], we choose the 8-bit grayscale images. The compression results are similar to RGB images. We also used the same 4 K UHD images (Test set) mentioned in [30] including 60 nature and city images with the resolution of 3840 × 2160.

3.2 Experimental results

The compression results shown in Table 1- Table 3 are expressed in bits per pixel (bpp), which is an absolute measure of compression ratio. It represents the average number of bits needed to encode the pixel information from an image. Bpp is calculated by dividing the compressed image size by the number of pixels in eq. (8). A smaller value of bpp denotes better compression performance.

$$\text{Bits Per Pixel} = \frac{\text{Number of Bits}}{\text{Number of Pixels}} \tag{8}$$

The six state-of-the-art methods, Huffman coding, arithmetic coding, JPEG 2000, JPEG-LS, 7-Zip lossless compression [1] and the proposed algorithm were applied to all the three image sets. Huffman coding and arithmetic coding both are well-recognized lossless entropy coding algorithms. JPEG 2000 and JPEG-LS are effective lossless/near-lossless image compression standards. 7-Zip is a new file archiver with a high compression ratio. We chose LZMA2 as a compression method in the 7-Zip program.

The best results in the tables are highlighted in bold type. As shown in Table 1, our approach outperforms the state-of-the-art ones in most cases. It can be noticed that the average result of our algorithm is much better than Huffman coding and arithmetic coding. It also obtains a 10.25% improvement compared to JPEG 2000, 6.22% compared to JPEG-LS and 16.39% compared to 7-Zip. We find that our algorithm suffers poor compression performance when it compresses simple images with fewer textures, such as "circles", "crosses", "slope" and "squares" in Table 1. In contrast, our method performs much better for images with more

Table 1 Compression results in bpp of the Waterloo image set

Images	Resolution	Huffman	Arithmetic	J2K	JLS	7-Zip	Proposed
bird	256×256	6.8016	6.7747	3.1390	3.4712	4.2333	**2.8630**
bridge	256×256	7.6937	7.6689	5.9086	5.7904	6.3164	**4.9043**
circles	256×256	1.8484	1.7811	1.2627	0.1526	**0.1136**	1.3375
crosses	256×256	1.0001	0.1879	1.4285	0.3855	**0.1786**	2.0281
slope	256×256	7.5411	7.5177	**1.0643**	1.5713	1.6929	1.6449
squares	256×256	1.3517	1.0776	0.2505	0.0771	**0.0500**	0.6989
boat	512×512	7.1468	7.1238	4.1005	4.2498	5.2881	**3.5184**
library	464×352	5.8704	5.8490	5.8350	5.1011	**4.2544**	5.1664
goldhill2	512×512	7.4970	7.4779	4.6544	4.7116	5.5985	**3.8035**
lena2	512×512	7.4683	7.4456	4.0166	4.2437	5.5190	**3.2545**
mandrill	512×512	7.3804	7.3580	6.0232	6.0365	6.3869	**5.0441**
peppers	512×512	7.5951	7.5716	4.4042	4.4887	5.5480	**3.5105**
Average	–	5.7662	5.6528	3.5073	3.3566	3.7650	**3.1478**

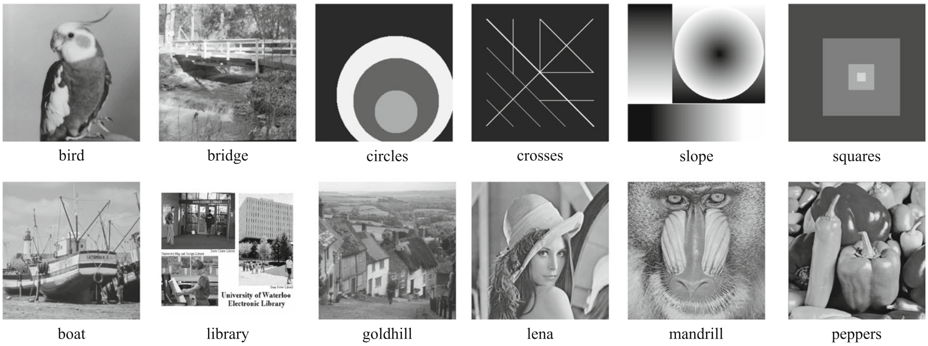

Fig. 6 12 different images from the waterloo image compression benchmark

complex texture and higher resolution. This shows that our method has better applicability for high-resolution and high-precision images.

The Paq8px167 + CM method also uses a simple model for contexts to predict the bits of the pixels, which is similar to our approach. In addition to the 5 classic methods above, therefore, we also compared our method with the Paq8px167 + CM method in Table 2. The

Table 2 Compression results in bpp of the new test images

Images	Resolution	Huffman	Arithmetic	J2K	JLS	7-Zip	Paq8px167 + CM	Proposed
artificial	3720 × 2048	6.3597	6.3342	1.0639	0.7976	0.7420	**0.3186**	0.8495
big_building	7216 × 5412	7.5308	7.5018	3.1071	3.5921	4.7640	3.1216	**2.4764**
big_tree	6088 × 4550	7.1974	7.1570	3.4121	3.7322	4.9004	3.3803	**2.3805**
Bridge	2749 × 4049	7.4571	7.4401	3.8934	4.1478	5.2383	3.7953	**2.8366**
cathedral	2000 × 3008	6.7662	6.7312	3.2557	3.5699	4.6852	3.1519	**2.3169**
Deer	4043 × 2641	6.0966	6.0709	4.4169	4.6592	5.1676	4.1750	**3.4039**
Fireworks	3136 × 2352	3.7693	3.7581	1.1640	1.4652	1.9581	1.2325	**0.8973**
flower_foveon	2268 × 1512	6.8392	6.8202	1.2456	2.0381	2.9506	1.6943	**0.9989**
Hdr	3072 × 2048	6.8746	6.8496	1.4095	2.1752	3.1322	1.8327	**1.1066**
leaves_iso_200	3008 × 2000	7.3305	7.2971	3.5532	3.8195	5.0657	4.0473	**2.7515**
leaves_iso_1600	3008 × 2000	7.4161	7.3814	4.3849	4.4863	5.6010	**3.2130**	3.3135
nightshot_iso_100	3136 × 2352	6.6007	6.5670	1.4065	2.1295	3.0175	1.7805	**1.1059**
nightshot_iso_1600	3136 × 2352	6.5108	6.4876	3.7366	3.9712	4.7706	3.6272	**2.6608**
spider_web	4256 × 2848	7.5980	7.5681	1.0361	1.7655	2.7854	1.3502	**0.7760**
zone_plate	3000 × 2000	7.7633	7.7269	4.4573	7.4293	1.5277	**0.1257**	5.9328
Average	–	6.8074	6.7794	2.7695	3.3186	3.7537	2.4564	**2.2538**

Table 3 Compression results in bpp of 60 4 K UHD images

	Huffman	Arithmetic	J2K	JLS	7-Zip	LocoNN	Proposed
Average	7.5796	7.5493	2.5462	3.0334	3.8735	2.8682	**2.0952**

Paq8px167 + CM method combines Paq8px167 with contextual memory and is proved having better compression performance than the state-of-the-art compression program PAQ8PX [8]. The images illustrated in Table 2 have higher resolution than those in Table 1, but in this case, our method still performs better. Table 2 shows that the average bpp of the proposed algorithm exceeds Huffman coding, arithmetic coding, JPEG 2000, JPEG-LS, 7-Zip, and Paq8px167 + CM by 66.89%, 66.76%, 18.62%, 32.09%, 39.96% and 8.25%, respectively.

Table 3 shows the compression results comparison of the 5 methods and the CNN-based lossless compression using LOCO-I predictor, called LocoNN [30], as well as the proposed algorithm. The compression results are the average bpp of a number of 60 4 K UHD (3840 × 2160) images. It can be noticed that our method outperforms the others by the bpp decreases of 72.36% compared to Huffman, 72.25% compared to arithmetic, 17.71% and 30.93% compared to JPEG 2000 and JPEG-LS, 45.91% compared to 7-Zip and 26.95% compared to LocoNN. Although a deep neural network is exploited in LocoNN, the complex algorithm structure may lead to poor compression results. As a result, the algorithm in this paper still achieves better compression performance.

The resolutions of images from Tables 1, 2 and 3 are gradually increasing, but the average bpps of our algorithm in three Tables (3.1478, 2.2538 and 2.0952) are decreasing. This indicates that the proposed method has better performance when it comes to high-resolution images.

The result shown in Table 4 is the running time comparison of the six methods compressing all the 12 images in Fig. 6. We can see that JPEG 2000, JPEG-LS and 7-Zip can compress images very fast, because we use well-packaged software of them in the experiments. Our algorithm is implemented in MATLAB (R2019b) platform, so it is not as fast as the three methods above in computing. We also use MATLAB to implement Huffman coding and arithmetic coding. They take much more compressing time than our method. Arithmetic coding takes so much time that is virtually unavailable for dynamic compression. Therefore, our method is moderate in speed, and we will further reduce the complexity of the algorithm in the future by improving our algorithm code structure in MATLAB or implementing it in other programming languages, such as Python and C++.

According to Shannon's theorem, data can be compressed based on the probability of it. Entropy determines the achievable compression ratio. The smaller image entropy value is, the greater compression result can be achieved. In order to demonstrate entropy performance of the proposed method, we compare the entropy of five schemes, i.e., DPCM-5/3, NP-5/3, DPCM-5/3-CP, the proposed one (NP-5/3-CP), and the DPCM-Haar method presented in [5]. We use the same images as presented in [5], which are well-known standard test images. Symbol 5/3 means 5/3 wavelet transform, NP means the novel predictor presented in this paper, and CP represents the IWT coefficients processing mentioned above.

Table 5 shows that the average entropy value of our method drops to 3.6074 from 7.3671 of the original one performing the best in entropy reduction. The entropy result of NP-5/3 is smaller than DPCM-5/3, and the result of NP-5/3-CP is smaller than NP-5/3. As a result, the

Table 4 Compression running time comparison of images in Fig. 6

	Huffman	Arithmetic	J2K	JLS	7-Zip	Proposed
Time(s)	12	150	0.6	0.1	0.01	2

new predictor (NP) and IWT coefficients processing step (CP) proposed in this paper can produce a lower entropy value and achieve a better compression ratio.

We use MSE to measure whether the image decompression recovery is complete. The MSE values of the test images are zero, proving that the proposed compression algorithm is indeed lossless.

4 Conclusions and future work

In this paper, we present an improved lossless image compression scheme combining lossless linear prediction, the LeGall5/3 integer wavelet transforms with coefficients processing and Huffman coding. A new hybrid transform exploiting a new prediction template and a coefficient processing of IWT is the main contribution of this algorithm. Due to the great correlation between the adjacent pixels of the original image, there is a large compression space. The two-dimensional linear prediction model in this paper greatly reduces the redundancy between the pixels, so that the error values are concentrated near 0 as much as possible, and the complexity of the data is reduced. The obtained predicted data entropy is greatly less than the original image entropy. After the LeGall5/3 integer wavelet transforms, zero and negative values are removed from the wavelet coefficients, making achievable compression ratio higher. In general, the proposed algorithm increases the compressible space by reducing the redundancy of the original image.

The experimental results show that the proposed algorithm outperforms other state-of-the-art algorithms as well as the methods presented recently. When images become more complex in higher resolution, the performance of our method becomes better, and the compression speed is acceptable. Our method mainly focuses on natural images with 256 colors in the present study. In the future, we will modify the method to work with a wider range of image types like color-indexed images with a small color palette. The complexity of the algorithm is expected to further reduced to optimize the compression speed. Besides, we will also try to propose a CNN-based lossless compression method in future work.

Table 5 Entropy comparison of five schemes

Images	Original	DPCM-5/3	NP-5/3	DPCM-5/3-CP	Proposed (NP-5/3-CP)	DPCM-Haar
Cameraman	7.0480	3.0437	2.7150	2.8946	2.7453	3.5825
Baboon	7.2925	5.3930	5.1499	4.0042	3.7507	6.3424
Lena	7.4455	4.2657	4.2145	3.7492	3.6279	4.7555
Peppers	7.5715	4.3380	4.1961	3.9460	3.8184	4.9787
Goldhill	7.4778	4.6569	4.5858	4.1645	4.0945	5.5163
Average	**7.3671**	**4.3395**	**4.1723**	**3.7517**	**3.6074**	**5.0351**

Acknowledgments This work was supported in part by the National Natural Science Foundation of China, under Grants 62020106011, 62071287, and 62001279.

Code availability Not applicable

Authors' contributions Not applicable

Funding This work was supported in part by the National Natural Science Foundation of China, under Grants 62020106011, 62071287, and 62001279.

Data availability Not applicable

Declarations

Conflicts of interest/competing interests Not applicable

References

1. 7-Zip (2021) https://www.7-zip.org/. Accessed 22 Feb. 2021
2. Oord Avd, Kalchbrenner N, Kavukcuoglu K (2016) Pixel Recurrent Neural Networks. Paper presented at the Proceedings of the 33rd International Conference on Machine Learning, 19 Aug.
3. Avramović A, Savić S (2011) Lossless predictive compression of medical images*. Serbian Journal of Electrical Engineering 8(1):27–36
4. Ayyoubzadeh SM, Wu X (2020) Lossless compression of mosaic images with convolutional neural network prediction. ArXiv abs/2001.10484
5. Azman NAN, Ali S, Rashid RA, Saparudin FA, Sarijari MA (2019) A hybrid predictive technique for lossless image compression. Bulletin of electrical engineering and informatics 8 (4):1289-1296. Doi:https://doi.org/10.11591/eei.v8i4.1612
6. Christopoulos C, Skodras A, Ebrahimi T (2000) The JPEG2000 still image coding system: an overview. IEEE Trans Consum Electron 46(4):1103–1127. https://doi.org/10.1109/30.920468
7. Ding J, Wang I (2016) Improved frequency table adjusting algorithms for context-based adaptive lossless image coding. In: 2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), pp 1–2. doi:https://doi.org/10.1109/ICCE-TW.2016.7521049
8. Dorobanţiu B (2019) Improving lossless image compression with contextual memory. Appl Sci 9(13). https://doi.org/10.3390/app9132681
9. Fawcett R (1996) Combination coding: a new entropy coding technique. In: Proceedings of Data Compression Conference - DCC '96, 31 March-3 April 1996. p 434. doi:https://doi.org/10.1109/DCC.1996.488366
10. Fleet PJV (2019) The JPEG2000 image compression standard. In: discrete wavelet transformations. 2 edn. John Wiley & Sons, pp 525-545. doi:https://doi.org/10.1002/9781119555414.ch12
11. Giudice O, Allegra D, Stanco F, Grasso G, Battiato S A (2018) Fast Palette Reordering Technique Based on GPU-Optimized Genetic Algorithms. In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp 1138–1142. doi:https://doi.org/10.1109/ICIP.2018.8451221
12. Golomb S (1966) Run-length encodings (Corresp.). IEEE Trans Inf Theory 12(3):399–401. https://doi.org/10.1109/TIT.1966.1053907
13. Hassen W, Amiri H (2013) The 5/3 and 9/7 Wavelet Filters Study in a Sub-bands Image Coding. In: 2013 7th IEEE International Conference on e-Learning in Industrial Electronics (ICELIE), pp 150–154. doi:https://doi.org/10.1109/ICELIE.2013.6701290
14. Hussain AJ, Al-Fayadh A, Radi N (2018) Image compression techniques: a survey in lossless and lossy algorithms. Neurocomputing 300:44–69. https://doi.org/10.1016/j.neucom.2018.02.094
15. Image Repository of the University of Waterloo (2021) http://links.uwaterloo.ca/Repository.html. Accessed 22 Feb. 2021
16. Jain C, Chaudhary V, Jain K, Karsoliya S (2011) Performance Analysis of Integer Wavelet Transform for Image Compression. In: 2011 3rd International Conference on Electronics Computer Technology, pp 244–246. doi:https://doi.org/10.1109/ICECTECH.2011.5941746

17. Jain P, Jain A, Agrawal C (2013) Effective dictionary based data compression and pattern searching in dictionary based compressed data. In: 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), pp 1–6. doi:https://doi.org/10.1109/ICCCNT.2013.6726570
18. Khandwani FI, Ajmire PE (2018) A survey of lossless image compression techniques. International Journal of Electrical Electronics & Computer Science Engineering 5(1):39–42
19. Kitanovski V, Kseneman M, Gleich D, Taskovski D (2008) Adaptive lifting integer wavelet transform for lossless image compression. In: 2008 15th International Conference on Systems, Signals and Image Processing, pp 105–108. doi:https://doi.org/10.1109/IWSSIP.2008.4604378
20. Kumar V, Sharma S (2017) Lossless image compression through Huffman coding technique and its application in image processing using MATLAB. International journal of soft computing and engineering (IJSCE):10-13
21. Kumar RN, Jagadale BN, Bhat JS (2019) A lossless image compression algorithm using wavelets and fractional Fourier transform. SN Applied Sciences 1(3):266. https://doi.org/10.1007/s42452-019-0276-z
22. Mentzer F, Agustsson E, Tschannen M, Timofte R, Gool LV (2020) Practical full resolution learned lossless image compression.
23. Oswal S, Singh A, Kumari K (2016) Deflate compression algorithm. International Journal of Engineering Research and General Science 4(1):430–436
24. Pinho AJ, Neves AJ (2004) A survey on palette reordering methods for improving the compression of color-indexed images. IEEE Trans Image Process 13(11):1411–1418. https://doi.org/10.1109/tip.2004.836168
25. Rahman MA, Hamada M (2019) Lossless image compression techniques: a state-of-the-art survey. Symmetry 11(10):1274–1296. https://doi.org/10.3390/sym11101274
26. Rahman MA, Rabbi MMF, Rahman MM, Islam MM, Islam MR (2018) Histogram modification based lossy image compression scheme using Huffman coding. In: 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT), pp 279–284. doi:https://doi.org/10.1109/CEEICT.2018.8628092
27. Reed S, Oord Avd, Kalchbrenner N, Gómez S, Wang Z, Belov D, Freitas Nd (2017) Parallel multiscale autoregressive density estimation. ICML 2017
28. Salimans T, Karpathy A, Chen X, Kingma DP (2017) PixelCNN++: improving the PixelCNN with discretized logistic mixture likelihood and other modifications. Paper presented at the ICLR 2017, 19 Jan
29. Savakis AE (2000) Evaluation of Lossless Compression Methods for Grayscale Document Images. In: Proceedings 2000 International Conference on Image Processing, pp 136–139. doi:https://doi.org/10.1109/ICIP.2000.900913
30. Schiopu I, Liu Y, Munteanu A (2018) CNN-based Prediction for Lossless Coding of Photographic Images. In: 2018 Picture Coding Symposium (PCS), pp 16–20. doi:https://doi.org/10.1109/PCS.2018.8456311
31. Sharma K, Gupta K (2017) Lossless data compression techniques and their performance. In: 2017 International Conference on Computing, Communication and Automation (ICCCA):256–261. https://doi.org/10.1109/CCAA.2017.8229810
32. Sheikh S, Narayanan A (2019) An Efficient Palette Reordering For Lossless Compression of Color Indexed Images. In: 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), pp 1671–1673. doi:https://doi.org/10.1109/ICICICT46008.2019.8993282
33. Shrikhande RN, Bairagi VK (2014) Image Compression Using Calic. In: 2014 International Conference on Advances in Communication and Computing Technologies (ICACACT 2014), pp 1–4. doi:https://doi.org/10.1109/EIC.2015.7230725
34. Sun Y-k A (2004) Two-dimensional Lifting Scheme of Integer Wavelet Transform for Lossless Image Compression. In: 2004 International Conference on Image Processing, pp 497–500 doi:https://doi.org/10.1109/ICIP.2004.1418799
35. Sweldens W (1998) The lifting scheme: a new philosophy in Biorthogonal wavelet constructions. Proceedings of SPIE-The International Society for Optical Engineering 2569(1):68–79
36. The New Test Images-Image Compression Benchmark (2021) http://imagecompression.info/test_images/. Accessed 22 Feb. 2021
37. Weinberger MJ, Seroussi G, Sapiro G (2000) The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS. IEEE Trans Image Process 9(8):1309–1324. https://doi.org/10.1109/83.855427
38. Wu X, Memon N (1997) Context-based, adaptive, lossless image coding. IEEE Trans Commun 45(4):437–444. https://doi.org/10.1109/26.585919
39. Y-l Z, X-p F, S-q L, Z-y X (2010) Improved LZW algorithm of lossless data compression for WSN. In: 2010 3rd International Conference on Computer Science and Information Technology:523–527. https://doi.org/10.1109/ICCSIT.2010.5563620