
This is the author’s final accepted version.

There may be differences between this version and the published version. You are advised to consult the publisher’s version if you wish to cite from it.

http://eprints.gla.ac.uk/257354/

Deposited on: 07 December 2021

Enlighten – Research publications by members of the University of Glasgow
http://eprints.gla.ac.uk
A Low-complexity Lossless Image Compression for Small Spacescrafts’ On-board Computers

Hieu V. Dang¹, Son C. Chu², Nhung T. T. Nguyen¹, and Can C. Chu¹

¹ Department of Communications Engineering
University of Transport and Communications, Hanoi, Vietnam
² Faculty of Art, Computing, Engineering and Science
Sheffield Hallam University, Sheffield, United Kingdom

Abstract—In this paper, we present a novel and low-complexity lossless compression for gray-scale images. The gray-scale image is first separated into bit-planes. These bit-planes are then performed a binary wavelet transform (BWT) to obtain an efficient representation for compression. The BWT bits of significant bit-planes are then encoded by the run-length coder that uses Golomb-Rice codes for run-encoding. The experimental results show that the algorithm obtained efficiency in image compression, and low-complexity in implementation that is highly applicable for image compression systems on small spacecraft’s on-board computers.

Index Terms—Lossless image coding; bit-plane coding; binary wavelet transform; Golomb-Rice coding; small spacecrafts’ on-board computers;

I. INTRODUCTION

Small satellites are recently increasing interest all over the world for their attractive applications. An advantage of small satellites is the fast and low-cost development, which makes them a suitable platform for evaluating and demonstrating rapid new technologies in space. Because of the limitation of available size, mass, and power of small satellites, demanding on high computational capabilities on small satellites is a challenge to engineers. The future need for small satellites is to process high resolution payload data (e.g., imaging payload), and complex control algorithms.

Satellite imaging payloads mostly operate a store-and-forward mechanism, whereby captured images are stored on board and transmitted to ground stations later on [1]. With the demanding of high spatial resolution imaging, space missions are faced with the complexity of processing and conveying an extensive amount of imaging data. Thus image compression becomes an important procedure in payload processing on-board computers of small satellites. Image compression reimburses for the limited on-board computing resources, such as mass memory and downlink’s bandwidth.

Image compression tries to exploit and remove redundancies in image to obtain a high compression ratio and an acceptable quality of the reconstructed image. There are different classes of redundancies in an image, such as spatial, statistical, and human vision redundancies. Image compression techniques can be classified into two classes, lossless and lossy image compressions. In lossless compression, the reconstructed image is identical to the original one (there is no loss of information). In an opposite manner, lossy image compression methods reconstruct the image with an information lost. Lossless image compression is required for applications that cannot tolerate any degradation of original image. For instance, satellite images or geographical map images, where it cannot be tolerated by distortion caused by compression techniques.

One approach for exploiting efficiently spatial correlations for compression is to decompose the image into a set of binary layers (bit-planes), and then compress these layers by a binary image compression technique [2], [3], [4], [5]. The decompression is an inverse process of the compression, where the compressed file is decompressed into a set of layers which are then combined back into the gray-scale image. The less significant bit-planes are typically difficult to predict the structures to be compressed well. This is because bit-plane separation destroys the gray-level correlations of the original image. Thus, single lossless coding methods such as dictionary-based, run-length codings are not efficient to encode these insignificant bit-planes.

Recently, wavelet transforms have been applied to reduce the entropy of the data source for lossless image compressions [6], [7], [8]. Multiresolution image representation with high coding efficiency are the most attractive attributes of wavelet-based coding methods. The wavelet transforms are almost for real-valued and complex-valued functions (i.e., the data to be analyzed, the basis functions, and the arithmetic operators are in the real or complex fields) [9]. These transforms have a degree of computational complexity. Swanson and Tewfik [9] have introduced the theory of binary wavelet transform (BWT) for binary images over the finite Galois field of order 2, GF(2). BWT shares many of the important characteristics of the real wavelet transform. Furthermore, this BWT has several distinct advantages over the real wavelet transforms, including: (i) the entire decomposition process is performed in GF(2), which means that the intermediate and transformed data produced by BWT are binary (This leads to no quantization effects introduced and the decomposition is completely invertible); (ii) the algorithm is extremely fast and much simpler since the data remains in GF(2) and the transform uses modulo-2 arithmetic operators which can be performed using simple Boolean operations.

In this paper, we present a lossless image compression
using bit-plane coding, BWT, and run-length/Golomb-Rice codes. The gray-scale image is first decomposed into bitplanes by bit-plane separation methods. These bit-planes are then sequentially performed BWT. The binary wavelet bits are scanned and then applied run-length coding with Golomb-Rice codes to obtain the compressed bit-stream. The proposed method has some important advantages listed as follows.

1. It is a reversible coding method, in which the reconstructed image is identical to the original one.
2. The algorithm is extremely fast since all the data processing flow are in binary computation in the GF(2).
3. The algorithm is simple and applicable for the simple implementations on microprocessors.

The rest of the paper is organized as follows. Sec. II gives a background on bit-plane coding, BWT, and Golomb-Rice codes. Sec. III describes the proposed compression method. Sec. IV discusses experimental results. Finally, the paper ends with the conclusions.

II. BACKGROUNDs

A. Bit-plane Coding

Bit-plane coding is a technique whereby a group of bits is divided into subgroups so that some of the subgroups can be summarily described [2]. Since data is commonly stored in a binary format in most electronic computing devices, one natural approach to implement an embedded coding system is through sequential bit-plane coding, where the input data are sequentially scanned and coded by bit-planes, usually from the most significant to the least, to generate the compressed bit-stream [3].

The foremost step in bit-plane coding is data decomposition where data is decomposed into different bit-planes for later encoding. There are three common decomposition methods: (i) binary-coded separation (BCS); (ii) gray-coded separation (GCS); and (iii) prediction-error separation (PES). In this work, we consider the BCS and GCS for bit-plane coding.

The BCS is a straightforward bit-plane separation. With a gray-scale image, pixels have values varying from 0 to 255, which can be represented by 8 bits binary data from the most significant bit MSB to the least significant bit LSB. These binary data is separated into 8 bit-planes from MSB bit-plane which contains MSB bits of all pixels to LSB bit-plane which correspondingly contains LSB bits of pixels. The eight bit-planes of gray-scale image Cameraman with the size of 256-by-256 are displayed in Fig. 1. The main disadvantage of BPS is that pixels which differ by 1 or 2 in decimal values differ in many bit positions in binary values.

The second separation method is a gray-code separation, in which the pixel intensities are represented by gray codes so that the change of pixel value by +1 or -1 causes the change of only one bit in the corresponding bit-planes. The gray-codes can be drawn by converting a binary number to a gray number. Thus, the gray-coded separation can be implemented from the binary-coded separation as follows. The gray-scale pixels are first represented by binary codes. The binary codes are then converted into gray codes. The bit-planes are created as the same procedure of BCS that include eight bit-planes from MSB to LSB bit-planes. An example of gray-coded separation for the gray image Cameraman is shown in Fig. 1.

B. Binary Wavelet Transform

1) Binary Field Transform: A binary field, also called the Galois Field of order 2 (GF(2)), has only two symbols 0 and 1. Operations of addition, subtraction, multiplication, and division are defined over these two symbols only. Real field transforms can be applied to the binary field, but it is complex in computation. To overcome this difficulty, the binary field transform (BFT) have been proposed by Swanson and Tewfik [9]. For finite sequences, the BFT takes the form of a square symmetric matrix. The construction of the BFT matrix and its inverse is as follows.

Let us define

\[ B_2 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \]

and

\[ B_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \]

For \( N \geq 6 \) and even, the BFT matrix \( B_N \) is constructed by

\[ B_N = \begin{bmatrix} B_{ul}^N & B_{ur}^{N/2} \\ B_{rl}^N & B_{rr}^{N/2} \end{bmatrix} \]

The upper-left submatrix with the size of \((N-2) \times (N-2)\) is given by

\[ B_{ul}^N = \begin{bmatrix} 1_{8x2} & 1_{8x(N-4)} \\ 1_{8(N-4)x2} & \overline{B}_{N-4} \end{bmatrix} \]

where \( 1_{8xM} \) is an NxM matrix of number 1. Matrix \( B_{N-4} \) is the result of applying the logical-not operation to each element of the BFT matrix \( B_{N-4} \).

The upper-right submatrix \( B_{ur}^{N/2} \) with the size of \((N-2)x2\) is defined by

\[ B_{ur}^{N/2} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 1 & 1 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \]

The lower-left submatrix \( B_{rl}^N \) with the size of \(2x(N-2)\) is defined as the transpose of \( B_{ur}^{N/2} \), given by

\[ B_{rl}^N = B_{ur}^{N/2} \]

For example, 8x8 BFT matrix is constructed as follow.

\[ B_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \]
Since \( \det(B_N) = 1 \) for all \( N \geq 2 \), \( B_N \) is invertible over the binary field. The inverse \( B_N^{-1} \) can be evaluated using a simple recursive formula for \( N \geq 6 \) as follows.

\[
B_N^{-1} = \begin{bmatrix}
A(N-2)x(N-2) & C(N-2)x2 \\
C^T(N-2)x2 & D_{2x2}
\end{bmatrix}
\]

where

\[
A(N-2)x(N-2) = \begin{bmatrix}
1 & 0 & 0 & \ldots & 0 & 0 \\
0 & 1 & 1 & 0 & \ldots & 0 & 1 \\
0 & 1 & & & & & \\
0 & 0 & & & & & \\
\vdots & & & & & & \\
0 & 0 & & & & & \\
1 & 1 \\
\end{bmatrix}
\]

\[
C(N-2)x2 = \begin{bmatrix}
1 & 0 \\
1 & 0 \\
0 & 0 \\
\vdots & \\
0 & 0
\end{bmatrix}
\]

\[
D_{2x2} = \begin{bmatrix}
1 & 1 \\
1 & 1
\end{bmatrix}
\]

For the 8x8 BFT, the inverse BFT matrix is

\[
B_8^{-1} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{bmatrix}
\]

The filter BFT (FBFT) of a vector \( x \) is performed by

\[
\hat{X} = XB_N
\]

2) Binary Wavelet Transform: The theory of BWT is developed from the principle of BFT and parallels with the theory of wavelet transform developed over the real field. The construction of two-band discrete orthonormal binary wavelets as equivalent to the design of a two-band perfect reconstruction filter bank with added vanishing moments as shown in Fig. 2.

A filter bank is often cascaded with one or more additional filter banks to provide further resolutions of the input signal.

To guarantee that the binary multiresolution decomposition inherits the important characteristics of the real wavelet decomposition, and still be able to reconstruct the original signal perfectly, the filters must satisfy three constraints: (i) the bandwidth constraint that restrict the bandwidths of the low-pass and high-pass filters to guarantee that no information is lost after the outputs of the two filters are downsampled by a factor of 2; (ii) the vanishing moment constraint to guarantee that the BWT of slowly varying binary sequences are very sparse; and (iii) the perfect reconstruction constraint to guarantee that the BWT is invertible. From the above three constraints, the binary filters for BWT are designed as follows.

Suppose the \( N \)-tap low-pass filter \( l \) and the \( N \)-tap high-pass filter \( h \) are formulated as \( l = [ l_0 \ l_1 \ \cdots \ l_{N-1} ] \) and \( h = [ h_0 \ h_1 \ \cdots \ h_{N-1} ] \). To satisfy the bandwidth, the vanishing moment, and the perfect reconstruction constraints, the filters \( l \) and \( h \) must follow the conditions given by [9]

\[
\begin{align*}
\sum_{i=0,\text{even}}^{N-2} l_i &= 0; \quad \text{and} \quad \sum_{i=1,\text{odd}}^{N-1} l_i = 1 \\
\sum_{i=0,\text{even}}^{N-2} h_i &= 1; \quad \text{and} \quad \sum_{i=1,\text{odd}}^{N-1} h_i = 1
\end{align*}
\]
From the designed filters \( l \) and \( h \), we then formulate two-circulant matrices \( L_2 = 2 - \text{circ}(l) \) and \( H_2 = 2 - \text{circ}(h) \). The BWT matrix \( T \) is then setup by
\[
T = \begin{bmatrix} L_2 \\ H_2 \end{bmatrix}
\] (16)

The 2-D BWT of an image \( F \) with the size \( N \times N \) is performed by
\[
\hat{F} = TFTT^T
\] (17)

The transform in Eq. (17) corresponds to passing the image \( F \) through a low-pass 2-D separable filter and three band-pass 2-D separable filters, and decimating by 2 in each direction as shown in Fig. 3.

The inverse BWT is given by
\[
\tilde{F} = T^{-1} \hat{F} \left( T^T \right)^{-1}
\] (18)

Since the BWT is perfect reconstruction, \( \tilde{F} \) and \( F \) are identical. To obtain multiresolution decomposition, we can successively applying the decimated output of each LL filter as the input to the next stage.

C. Golomb-Rice Codes

Golomb-Rice coders have been applied widely in image compression systems [10]. Golomb-Rice coders are optimal or nearly optimal for integer sources with two-sided geometric distributions, which approximate quite closely the distribution of uniformly quantized Laplacian sources [11]. The main advantage of Golomb-Rice codes is that the output codewords are easily computed for the corresponding input symbols by changing a single integer parameter \( k \), so that no explicit tables are actually required. This makes the computation of Golomb-Rice coders much faster than memory access based coding. The theory and implementation of Golomb-Rice coders are briefly introduced as follows.

For an integer number \( n \), the principle of Golomb-Rice coder with parameter \( k \) is defined by the encoding rule in Fig. 4 [12], [13], [11].

![Fig. 4. Golomb-Rice coding principle with parameter \( k \).](image)

The prefix \( q \) bits is also called the quotient part which consists of \( q \) unary code bits. The remainder \( r \) is the fixed-length code bits, \( k \) LSB of the number \( n \). For example, with \( k = 3 \), Golomb-Rice codes for number \( n = 1, 4, 8, 11, 16, 20 \) are shown in Table I.

<table>
<thead>
<tr>
<th>Input Value</th>
<th>Output Codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n )</td>
<td>( G(n,k) = 111 \ldots 110 b_{k-1} b_{k-2} \ldots b_2 )</td>
</tr>
</tbody>
</table>

![Fig. 5. The proposed lossless image compression.](image)

### III. PROPOSED ALGORITHMS

In this section, the proposed lossless compression of grayscale image is presented. The proposed technique is described in Fig. 5.

The proposed algorithm is described as follows. The grayscale image is first decomposed into bit-planes by using gray coded separation (GCS) method. The bit-planes are then decomposed in 3 levels by the binary wavelet transform (BWT) sequentially from the MSB bit-plane to the LSB bit-plane. The binary wavelet coefficients of each significant bit-planes (e.g., bit-planes MSB, 7, 6, and 5) are scanned to bit sequences for run-length/Golomb-Rice encoding with the parameter \( k = 3 \). The scanning procedure used in this work is shown in Fig. 7.
The decompression algorithm is the inverse procedure of the compression algorithm.

The scaling filters used for the 3-level BWT are the 8-tap low-pass filter \( I = [1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0] \), and the 8-tap high-pass filter \( h = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0] \). The BWT matrix \( T \) is then setup according to Eq. 16 by

\[
T = \begin{bmatrix}
1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\
\end{bmatrix}
\]  

(20)

The 3-level BWT decomposition of the bit-plane MSB of image Cameraman decomposed by gray-coded separation in Fig. 1 is shown in Fig. 6.

To measure the compactness of the binary image representation in 3-level BWT, we use the entropy function calculated by

\[
H(p) = -p \log_2(p) - (1 - p) \log_2(1 - p)
\]

where \( p \) is the probability of pixel value 1 in the binary bit-plane images. The measure takes values between 0 and 1. If the entropy is small, it indicates more efficient compression representation, and vice versa. The entropy results for four different input images are displayed in Tables II, III, IV, and V.

**TABLE II**  
ENTROPY RESULTS FOR INPUT IMAGE ‘CAMERAMAN’.

<table>
<thead>
<tr>
<th>Bit-planes</th>
<th>Original Entropy</th>
<th>BWT Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSB</td>
<td>0.9732</td>
<td>0.6784</td>
</tr>
<tr>
<td>7</td>
<td>0.8305</td>
<td>0.4898</td>
</tr>
<tr>
<td>6</td>
<td>0.9845</td>
<td>0.7502</td>
</tr>
<tr>
<td>5</td>
<td>0.9999</td>
<td>0.8811</td>
</tr>
<tr>
<td>4</td>
<td>0.9820</td>
<td>0.9093</td>
</tr>
<tr>
<td>3</td>
<td>0.9998</td>
<td>0.9515</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.9998</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

From the entropy results in Tables II, III, IV, and V, it can be seen that the entropy is significantly reduced at bit-planes MSB, 7, 6, and 5 after applying 3-level BWT. It indicates
that the 3-level BWT is an efficient representation for the compression of significant bit-planes. At the lower bit-planes, the increases of BWT entropy imply that the BWT become less efficient for the less significant bit-planes.

To evaluate the compression efficiency, we adopt the compression ratio (CR) and the reduction percentage (RP) given by

$$CR = \frac{V}{V^*}$$  \hspace{1cm} (22)

$$RP = \frac{V - V^*}{V} \times 100\%$$  \hspace{1cm} (23)

where $V$ is the data volume of the original signal in binary bits, and $V^*$ is the data volume of the encoded signal (i.e., length of the compressed bit-stream). The CR and RP results for the compressions of four different test images are described in Table VI.

The results in Table VI show that the proposed algorithm obtains a moderate-well compression effectiveness in lossless compression landscape.

V. CONCLUSIONS

We have studied efficient and low-complexity lossless compression methods in binary domain. The theory of binary wavelet transform with all computation in $GF(2)$ is simple and low-cost in hardware implementation. The Golomb-Rice coder is also studied for fast and low-cost implementation.

The proposed lossless compression for gray-scale image is presented. The experimental results show that the algorithm obtains good efficiency in compression and less complexity in implementation. The algorithm can be generalized for lossless compression of any real data and can be implemented on a small spacecraft’s on-board computer.

| TABLE III | ENTROPY RESULTS FOR INPUT IMAGE ‘CLOCK’.
| Bit-planes | Original Entropy | BWT Entropy |
| MSB 7 | 0.7064 | 0.4645 |
| 6 | 0.8616 | 0.5208 |
| 5 | 0.9964 | 0.7159 |
| 4 | 0.8245 | 0.7672 |
| 3 | 0.9569 | 0.8782 |
| 2 | 0.9820 | 0.9584 |
| 1 | 0.9954 | 0.9909 |
| 1 | 1 | 0.9995 |

| TABLE IV | ENTROPY RESULTS FOR INPUT IMAGE ‘LENA’.
| Bit-planes | Original Entropy | BWT Entropy |
| MSB 7 | 0.9994 | 0.6461 |
| 6 | 0.7653 | 0.6607 |
| 5 | 0.9988 | 0.7501 |
| 4 | 0.9947 | 0.8920 |
| 3 | 0.9999 | 0.9710 |
| 2 | 1 | 0.9952 |
| 1 | 1 | 0.9999 |
| 1 | 1 | 1 |

| TABLE V | ENTROPY RESULTS FOR INPUT IMAGE ‘HOUSE’.
| Bit-planes | Original Entropy | BWT Entropy |
| MSB 7 | 0.9994 | 0.7287 |
| 6 | 0.9974 | 0.5765 |
| 5 | 0.8660 | 0.7628 |
| 4 | 0.9794 | 0.9246 |
| 3 | 0.9666 | 0.9683 |
| 2 | 0.9999 | 0.9993 |
| 1 | 1 | 0.9998 |

| TABLE VI | CR AND RP RESULTS.
| Test Images | CR | RP |
| Cameraman | 1.2441 | 19.62 % |
| Clock | 1.3218 | 24.34 % |
| Lena | 1.2285 | 18.60 % |
| House | 1.4058 | 28.87 % |

REFERENCES