

Reversible Data Hiding Using the Companding Technique and Improved DE Method

Shaowei Weng · Yao Zhao · Jeng-Shyang Pan ·
Rongrong Ni

Received: 1 April 2007 / Revised: 8 September 2007
© Birkhäuser Boston 2008

Abstract A novel reversible data hiding scheme based on the companding technique and an improved difference expansion (DE) method is presented in this article. The companding technique is employed to embed payload into high frequency bands in the integer wavelet transform (IWT) domain. A refined method based on histogram modification is proposed as a remedy to the potential problem on overflow/underflow in pixel values after reverse IWT. To be truly reversible, additional information incurred by the histogram modification process is ingeniously embedded into the carrier image by using an improved DE method. A series of experiments is conducted to verify the feasibility and effectiveness of the proposed approach. As revealed in the experimental results, the proposed scheme outperforms existing approaches, in terms of embedding capacity and visual quality, as expected.

Keywords Reversible watermarking · Histogram shifting ·
The improved DE method

This work was supported in part by National Natural Science Foundation of China (Nos. 90604032, 60776794, 60702013), Specialized Research Fund for the Doctoral Program of Higher Education, Program for New Century Excellent Talents in University, Specialized Research Foundation of BJTU, 973 program (No. 2006CB303104), Natural Science Foundation of Beijing (No. 4073038).

S. Weng (✉) · Y. Zhao · R. Ni
Institute of Information Science, Beijing Jiaotong University, Beijing 100044,
People's Republic of China
e-mail: wswweiwei@126.com

Y. Zhao
e-mail: yzhao@bjtu.edu.cn

R. Ni
e-mail: rrni@bjtu.edu.cn

J.-S. Pan
Department of Electronic Engineering, Kaohsiung University of Applied Sciences, Kaohsiung,
Taiwan
e-mail: jspace@cc.kuas.edu.tw

1 Introduction

For some applications such as the fields of law enforcement, medical and military image systems, it is crucial to restore original images without any distortions. The watermarking techniques satisfying those requirements are referred to as reversible watermarking. Reversible watermarking is designed so that it can be completely removed to reconstruct the original image.

The concept of reversible watermark firstly appeared in the patent owned by Eastman Kodak [7]. Honsinger et al. utilized a robust spatial additive watermark combined with modulo addition to achieve reversible data embedding [7]. Fridrich et al. proposed another reversible watermarking scheme, which losslessly compressed high level bit-planes to make space for embedded data in the spatial domain [5]. Macq employed the patchwork algorithm and modulo addition to reversibly embed watermarking bits into host images [8]. Since the algorithms above aimed at authentication, the amount of hidden data was limited. The algorithms [1–4, 6, 9, 11] achieved a high hiding capacity. Goljan et al. proposed a two cycles flipping permutation to assign a watermarking bit in each pixel group [6]. Celik et al. presented a high capacity, reversible data embedding algorithm with low distortion by compressing quantization residues [2]. Tian presented a reversible data embedding approach based on expanding the pixel value difference between neighboring pixels which would not overflow or underflow after expansion [9]. Alattar used the difference expansion of triplets to embed a large amount of data into the grayscale images [1]. In the method proposed by Coltuc et al., an integer transform defined on pixel pairs was proposed to add the difference value of the pixel pair to one pixel and subtract the same value from the other [4]. Coltuc et al. generalized the basic idea for groups of an arbitrary number of pixels [3].

Xuan's method embedded data into high-frequency coefficients of an integer wavelet transform (IWT) with the companding technique, and utilized histogram modification as a preprocessing step to reduce the range of 8-bit grayscale to $[N, 255 - N]$ ($N > 0, N \in \mathbb{Z}$) in order to prevent overflow/underflow caused by the modification of wavelet coefficients [11]. The selection of parameter N was related to the threshold of the companding technique. The parameter N became larger as the threshold increased. Hence, the visual distortion caused by histogram modification was more perceptible, and the bitstream, produced by compressing a location map which identified histogram modification locations, also consumed a larger part of the hiding capacity. Otherwise, histogram modification modified all pixel values in the range $[0, N - 1] \cup [255 - N + 1, 255]$, but it didn't matter whether they really suffered overflow/underflow or not during embedding. Hence, histogram modification as a preprocessing step was not satisfactory.

To avoid the above problems, we interchange the order of histogram modification and IWT. In the IWT domain, the companding technique is exploited to just embed the payload into high-frequency coefficients without any histogram modification information, so the hiding capacity is considerably increased. Histogram modification is used after inverse IWT to make all overflowed or underflowed pixel values fall into the range of $[0, 255]$. In this paper, an improved difference expansion (DE) method is proposed to embed additional information incurred by histogram modification into marked images. Although the improved DE method introduces a reduction

of the peak signal-to-noise ratio (PSNR) value to the marked image, the overall PSNR value still exceeds that of Xuan’s method at the same rates.

The rest of the paper is organized as follows. In Sect. 2, Xuan’s method is reviewed. The embedding process and extracting process are presented in Sects. 3 and 4, respectively. The experimental results are shown in Sect. 5 and finally we conclude the paper in Sect. 6.

2 Xuan’s Method

Figure 1 from paper [11] shows the data embedding diagram of Xuan’s method. His method uses histogram modification as a preprocessing step to prevent overflow/underflow caused by the modification of wavelet coefficients. Histogram modification narrows down the range of 8-bit grayscale to $[N, 255 - N]$. As shown in Fig. 2, pixel values outside the range $[N, 255 - N]$ are shifted into it. To find those modified pixel values on the decoding side, histogram modification information called bookkeeping information needs to be recorded as a part of the embedded data.

Followed by histogram modification, wavelet CDF(2, 2) is performed on images after histogram modification in Xuan’s method. Table 1 shows the forward and inverse transform formula of CDF(2, 2) wavelet.

After IWT, Xuan’s method applies the companding technique to embed one bit into each high-frequency coefficient. Let C_h denote one coefficient of three high-frequency subbands LH, HL, HH . The main idea of the companding technique first proposed in paper [10] is depicted in the next subsection.

Fig. 1 Data embedding diagram

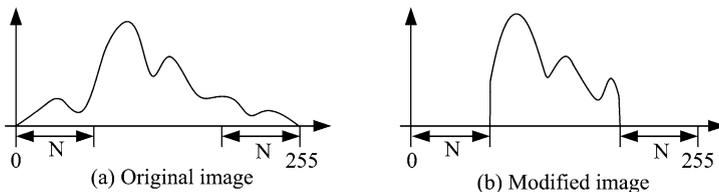
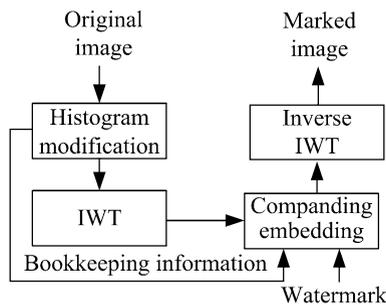
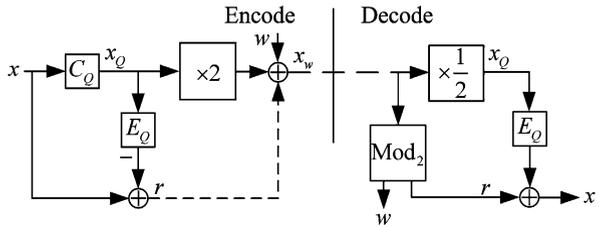


Fig. 2 Histogram modification in Xuan’s method

Table 1 CDF(2, 2) wavelet

| Forward transform | Inverse transform |
|---|---|
| Splitting: $s_i = x_{2i}$ $d_i = x_{2i+1}$ | Inverse primary lifting: $s_i = s_i - \lfloor \frac{d_{i-1} + d_i}{4} + \frac{1}{2} \rfloor$ |
| Dual lifting: $d_i = d_i - \lfloor \frac{s_i + s_{i+1}}{2} + \frac{1}{2} \rfloor$ | Inverse dual lifting: $d_i = d_i + \lfloor \frac{s_i + s_{i+1}}{2} + \frac{1}{2} \rfloor$ |
| Primary lifting: $s_i = s_i + \lfloor \frac{d_{i-1} + d_i}{4} + \frac{1}{2} \rfloor$ | Merging: $x_{2i} = s_i$ $x_{2i+1} = d_i$ |

Fig. 3 The framework of the companding technique for watermarking



2.1 Companding Technique

The companding technique contains a compression function C , an expansion function E and the bit-shift operation in this paper. For the original signal x , C and E have a relationship as follows:

$$E(C(x)) = x. \tag{1}$$

In the digital case, C_Q and E_Q respectively represent quantized versions of C and E , where Q denotes the quantization function. Obviously, for some signals, C_Q may map more than one input to the same output and E_Q will not reconstruct the original signal. So it causes the companding error, defined as

$$r = E_Q(C_Q(x)) - x \neq 0. \tag{2}$$

Hence, to retrieve original signal x , the companding error r needs to be losslessly compressed as a part of the embedded data. The framework of the companding technique for watermarking is illustrated in Fig. 3, where the \mathbf{Mod}_2 function represents the modulo-2 operation. Mathematically, it is equivalent to $\mathbf{Mod}_2(x_w) = x_w - 2 \times \lfloor \frac{x_w}{2} \rfloor$, where $\lfloor \cdot \rfloor$ rounds the value towards minus infinity.

The quantized compression function C_Q given below is adopted in Xuan’s method, namely

$$C_Q(x) = \begin{cases} x, & |x| < T_h, \\ \mathbf{sign}(x) \times (\lfloor \frac{|x| - T_h}{2} \rfloor + T_h), & |x| \geq T_h, \end{cases} \tag{3}$$

where T_h is a predefined threshold, the **sign** function returns a value indicating the sign of number x , and $|\cdot|$ represents the operation of returning the absolute value. As

the corresponding operation pair of the compression function, the expansion function is defined as

$$E_Q(x) = \begin{cases} x, & |x| < T_h, \\ \text{sign}(x) \times (2|x| - T_h), & |x| \geq T_h. \end{cases} \quad (4)$$

In (3) and (4), for $|x| < T_h$, x is equal to $E_Q(C_Q(x))$ and accordingly, the companding error r is zero. While for $|x| \geq T_h$, r is non-zero. However, the difference between x and $E_Q(C_Q(x))$ can not ensure r to be 1-bit binary number, so the analysis process below is applied to ensure $r \in \{0, 1\}$.

Equation (3) can be denoted by the following equation:

$$|x_Q| = \left\lfloor \left\lfloor \frac{|x| - T_h}{2} \right\rfloor + T_h \right\rfloor. \quad (5a)$$

If $|x| \geq T_h$ and $T_h > 0, T_h \in \mathbb{Z}$, then

$$|x_Q| - T_h = \left\lfloor \frac{|x| - T_h}{2} \right\rfloor. \quad (5b)$$

Let $y = |x| - T_h$, and let its binary representation be $b_7b_6 \cdots b_1r$, $b_i, r \in \{0, 1\}$, $1 \leq i \leq 7$. Then $\lfloor \frac{y}{2} \rfloor = b_7b_6 \cdots b_1 = \frac{y-r}{2}$. Substituting into (5b), we derive

$$r = |x| - (2 \times |x_Q| - T_h) = |x| - |E_Q(C_Q(x))|. \quad (6)$$

Hence, for $|C_h| \geq T_h$, the companding errors $r \in \{0, 1\}$ are collected as part of the embedded data. So the to-be-embedded data come from the following three parts: (1) watermark bits; (2) bookkeeping information; (3) the companding errors. For simplicity, the symbol w is utilized to denote one to-be-embedded bit and it is embedded into the least significant bit (LSB) of one bit left shifted version of $C_Q(C_h)$ as follows:

$$C_w = 2 \times C_Q(C_h) + w, \quad w \in \{0, 1\}, \quad (7)$$

where C_w is one marked coefficient. The data-hiding capacity in Xuan's method is given by

$$D = \|(|C_h| < T_h)\| - L_S, \quad (8)$$

where $\|\cdot\|$ is the cardinality of a set or the length of a sequence. L_S is the bit length of the bookkeeping information.

On the decoding side, first the embedded data is extracted as follows:

$$w = C_w - 2 \left\lfloor \frac{C_w}{2} \right\rfloor, \quad C_Q(C_h) = \left\lfloor \frac{C_w}{2} \right\rfloor. \quad (9)$$

At last, the original coefficient C_h is retrieved as follows:

$$C_h = \begin{cases} C_Q(C_h), & |C_Q(C_h)| < T_h, \\ \text{sign}(C_Q(C_h))((2|C_Q(C_h)| - T_h) + r), & |C_Q(C_h)| \geq T_h. \end{cases} \quad (10)$$

3 The Proposed Data Embedding Method

The histogram modification in Xuan's method modified all pixel values of the range $[0, N - 1] \cup [255 - N + 1, 255]$, but it didn't matter whether they really suffered overflow/underflow or not in the embedding process. Hence, the proposed method exchanges the order of histogram modification and IWT. The advantages are as follows:

- (1) Hiding capacity is increased with the PSNR value slightly increased. Since histogram modification is operated after inverse IWT, all high-frequency coefficients with amplitudes less than T_h are entirely utilized for carrying watermark bits without any bookkeeping information incurred by the histogram modification.
- (2) The overall PSNR is improved. Histogram modification as a postprocessing step just modifies those really overflowed/underflowed pixel values. As the experimental results show, the improved DE method used for the embedding of additional information incurred by the histogram modification process just causes the slight decline of PSNR value.

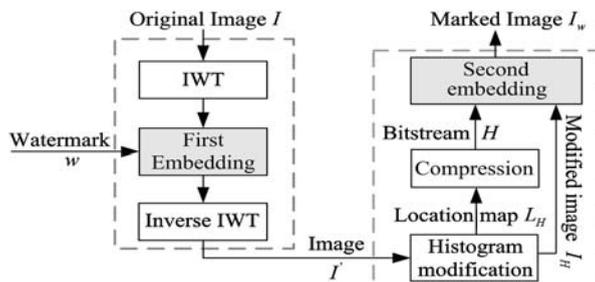
The main framework of the data embedding process is shown in Fig. 4. To compare with Xuan's method, we also utilize the CDF(2, 2) wavelet in this paper to be preformed on the original image I . Since histogram modification is operated after inverse IWT, the to-be-embedded data in the proposed method just include the following two parts: (1) watermark bits; (2) the companding errors. The detailed embedding process using the companding technique is discussed in Sect. 2. When the whole embedding process is finished, inverse IWT is performed.

3.1 Histogram Modification

After the inverse IWT, some pixel values in the image I' would fall outside the range of $[0, 255]$. For these pixel values, histogram modification is used to make them fall into the range of $[0, 255]$.

The detailed process of histogram modification is illustrated in Fig. 5. The minimum value less than zero and the maximum value larger than 255 are respectively found in the image I' . Next, the difference δ_1 between zero and the minimum value is added to all underflowed pixel values. Correspondingly, the difference δ_2 between the maximum value and 255 is subtracted from all overflowed pixel values. Hence, after modification, the overflow and underflow are prevented.

Fig. 4 The main framework of data embedding process



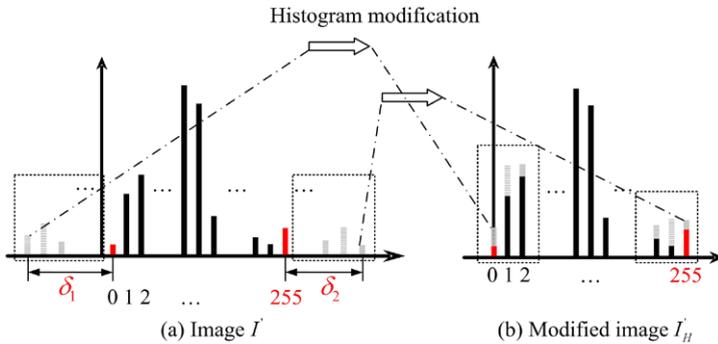


Fig. 5 Histogram modification in the proposed method

The 8-bit representations of δ_1 and δ_2 are concatenated to form a sub-bitstream \mathcal{H}_1 . To identify these modified pixel values, we create a binary location map L_H to record their positions. Then, it is losslessly compressed to form another sub-bitstream \mathcal{H}_2 . Finally, sub-bitstream \mathcal{H}_1 and sub-bitstream \mathcal{H}_2 are concatenated to form the overall bitstream \mathcal{H} , that is $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$.

3.2 Improved DE Method

After histogram modification, we propose the improved DE method to embed bitstream \mathcal{H} into the modified image I'_H . To clarify our approach, a brief discussion on the DE method is given first, followed by a detailed explanation of our method.

3.2.1 DE Method

For an 8-bit grayscale-valued pair (p_x, p_y) , average l and difference h are calculated via (11)

$$l = \left\lfloor \frac{p_x + p_y}{2} \right\rfloor, \quad h = p_x - p_y. \tag{11}$$

The inverse integer transform is as follows:

$$p_x = l + \left\lfloor \frac{h + 1}{2} \right\rfloor, \quad p_y = l - \left\lfloor \frac{h}{2} \right\rfloor. \tag{12}$$

To prevent possible overflows (>255) and underflows (<0), i.e., to restrict p_x, p_y in the range of $[0, 255]$, it is equivalent to have

$$|h| < \min(2(255 - l), 2l + 1), \tag{13}$$

h is left shifted by one bit and then one bit w_1 is appended into the LSB to get a new difference h_e . Mathematically, this is equivalent to

$$h_e = 2 \times h + w_1, \tag{14}$$

where $w_1 \in \{0, 1\}$. The reversible data embedding operation is called DE.

Each h is classified into one of the following three categories: expandable set, changeable set, and nonchangeable set.

If the expanded difference value h_e via (14) satisfies inequality (13), h is classified into the expandable set.

If the changed difference value, created by replacing the LSB of h (not in the expandable set) with 1-bit watermark information, satisfies inequality (13), h is classified into the changeable set.

The nonchangeable set contains the rest of h .

A location map is created by assigning a '1' to all differences in the expandable set and a '0' to the others. To ensure reversibility, the location map is losslessly compressed as a part of the embedded data. Hence, it typically consumes a large part of the available capacity. If a large part of the pixel pair can be correctly retrieved without the need of being recorded, then the embedding capacity can be considerably increased due to the large decrease in the length of the location map. This is the main idea of the improved DE method, which adopts a different classification method compared with Tian's method.

3.2.2 Classification Method

Similar to the twice-try structure proposed in [12], each h is classified into Class 1, Class 2 or Class 3 according to the following definition.

Definition 1 For all values of $w_1 \in \{0, 1\}$, if the expanded value h_e based on (14) does not satisfy inequality (13), then h belongs to Class 1.

For all values of $w_1, w_2 \in \{0, 1\}$, if h_e satisfies inequality (13), but h_{2e} given by (15) does not satisfy inequality (13), then the h belongs to Class 2.

For all values of $w_1, w_2 \in \{0, 1\}$, if both h_e and h_{2e} satisfy inequality (13), then h belongs to Class 3.

$$h_{2e} = 2 \times h_e + w_2. \quad (15)$$

From the above definition, h in Class 1 can not be expanded, while h in Class 2 can be expanded once and h in Class 3 can be expanded twice. That is to say, the new pixel pair computed by l and h_{2e} is still in the range of $[0, 255]$. Data is embedded into LSBs of h_e in Class 2 and Class 3. Hence, on the decoding side, h_e in Class 3 can be directly detected by once expansion without any additional information. So, it is unnecessary to record h in Class 3 during embedding.

To differentiate Class 1 from Class 2 on the decoding side, a location map is created by assigning '1' to h in Class 1 and a '0' to h in Class 2. This map's length just accounts for a small portion of all pixel pairs. After it is losslessly compressed, the compressed bit length is even smaller than that in the DE method. Thereby, the proposed method increases the hiding capacity.

h in Class 2 is further classified into Flipping Class 2 and non-Flipping Class 2. For each value of $k \in \{0, 1, 2, 3\}$, if

$$|4 \times h + k| > \min(2(255 - l), 2l - 1) \quad (16)$$

is true, h is classified into non-Flipping Class 2. Otherwise, h belongs to Flipping Class 2.

3.2.3 Embedding Process

The embedding process starts with selecting a certain number of pixel pairs (p_x, p_y) in I'_H according to a predetermined order which may serve as a security key.

A location map L_{DE} is created by assigning '1' to h in Class 1 and a '0' to h in Class 2. Then it is losslessly compressed by an arithmetic encoder into a bitstream \mathcal{L} with an end of message symbol at the end.

Bitstream \mathcal{L} is embedded into the LSBs of h_e in Class 3. Bitstream \mathcal{H} is embedded into the LSBs of the remaining h_e in Class 3 and all h_e in non-Flipping Class 2. Each h in Class 1 is kept intact. For h in the Flipping Class 2, if we embed a pseudo bit '1' into the LSB of h_e , h_{2e} does not satisfy inequality (13). Hence, the pseudo bit '1' is embedded. Otherwise, the pseudo bit '0' is embedded.

Finally, the inverse integer transform (12) is performed and the marked image I_W is produced. Note that the number of selected pixel pairs must exceed the length of bitstream $\mathcal{B} = \mathcal{L} \cup \mathcal{H}$.

4 Extraction Process

The extraction process is also composed of two stages corresponding to the embedding process. Stage 1 retrieves histogram modification information using the improved DE method. With the help of the extracted location map L_H in Stage 1, Stage 2 performs the inverse histogram modification and extracts the payload in the wavelet domain.

Stage 1. The same number of pixel pairs is selected according to the same order as in the embedding process. Next, the average values l_w and differences value h_w are calculated via (11). The classification process is illustrated in Fig. 6.

By identifying the end of message in the LSBs of all h_w in Class 3, the bits from the start until the end of message are decompressed by an arithmetic decoder to retrieve the location map L_{DE} . By L_{DE} , Class 1 is differentiated from Class 2. By dividing

Fig. 6 Classification structure

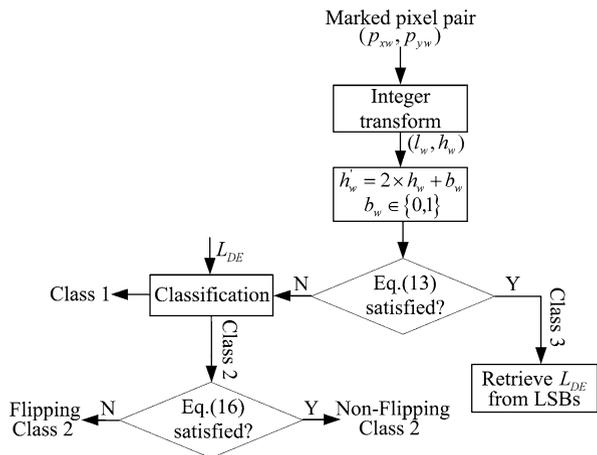


Table 2 Capacity vs. PSNR of watermarked 'Lena' image with size 256×256

| T_h | Gray-scale 'Lena' | | | |
|-------|-------------------|----------------|--------------|----------------|
| | Xuan' method | | Prop. method | |
| | PSNR (dB) | Payload (bits) | PSNR (dB) | Payload (bits) |
| 3 | 43.1883 | 32314 | 42.6594 | 34274 |
| 4 | 41.5988 | 35335 | 41.2267 | 37579 |
| 5 | 39.9812 | 35900 | 40.0413 | 39670 |
| 6 | 38.4073 | 35840 | 39.0972 | 41153 |
| 7 | 37.7566 | 37012 | 38.2804 | 42287 |
| 8 | 36.2854 | 36060 | 37.5618 | 43146 |
| 9 | 35.0100 | 35465 | 36.9728 | 43848 |
| 10 | 34.2812 | 35487 | 36.4197 | 44448 |
| 15 | 30.3686 | 32951 | 34.4915 | 46354 |
| 20 | | | 33.2406 | 47376 |

Table 3 Capacity vs. PSNR of watermarked 'Goldhill' image with size 512×512

| T_h | Gray-scale 'Goldhill' | | | |
|-------|-----------------------|----------------|--------------|----------------|
| | Xuan' method | | Prop. method | |
| | PSNR (dB) | Payload (bits) | PSNR (dB) | Payload (bits) |
| 3 | 42.0998 | 82503 | 42.0102 | 83843 |
| 4 | 40.1558 | 105105 | 40.1403 | 106740 |
| 5 | 38.6624 | 120812 | 38.7166 | 124813 |
| 6 | 37.5212 | 134059 | 37.6262 | 138867 |
| 7 | 36.4788 | 142913 | 36.7647 | 150115 |
| 10 | 34.7519 | 163242 | 35.0470 | 171273 |
| 15 | 32.8216 | 174489 | 33.5016 | 185657 |
| 20 | | | 32.6581 | 191077 |

Table 4 Capacity vs. PSNR of watermarked 'Build' image with size 512×512

| T_h | Gray-scale 'Build' | | | |
|-------|--------------------|----------------|--------------|----------------|
| | Xuan' method | | Prop. method | |
| | PSNR (dB) | Payload (bits) | PSNR (dB) | Payload (bits) |
| 3 | 37.7901 | 71208 | 42.5210 | 113661 |
| 4 | 34.1203 | 81930 | 40.6591 | 132718 |
| 5 | 31.3331 | 89554 | 39.2458 | 145977 |
| 6 | 30.4442 | 98893 | 38.1445 | 155499 |
| 10 | | | 35.2991 | 174160 |
| 15 | | | 33.1466 | 182742 |

Table 5 Capacity vs. PSNR of watermarked ‘Baboon’ image with size 512×512

| T_h | Gray-scale ‘Baboon’ | | | |
|-------|---------------------|----------------|--------------|----------------|
| | Xuan’s method | | Prop. method | |
| | PSNR (dB) | Payload (bits) | PSNR (dB) | Payload (bits) |
| 3 | 41.6243 | 48092 | 41.6299 | 48140 |
| 4 | 39.4091 | 64524 | 39.4063 | 64572 |
| 5 | 37.7249 | 79205 | 37.7198 | 79253 |
| 6 | 36.3844 | 91912 | 36.3836 | 91960 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 18 | 29.5987 | 161428 | 29.5061 | 161917 |
| 19 | 29.2771 | 162938 | 29.2299 | 164519 |
| 23 | 28.1734 | 169036 | 28.1965 | 172972 |



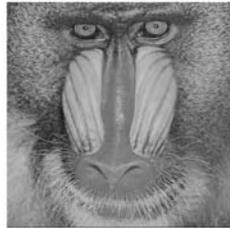
(a) ‘Lena’ image



(b) ‘Goldhill’ image



(c) ‘Build’ image

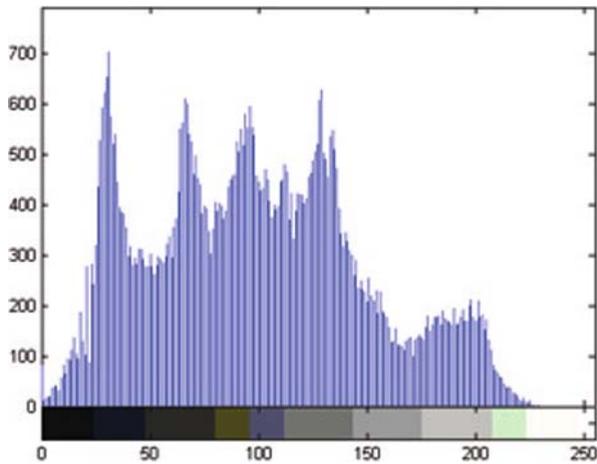


(d) ‘Baboon’ image

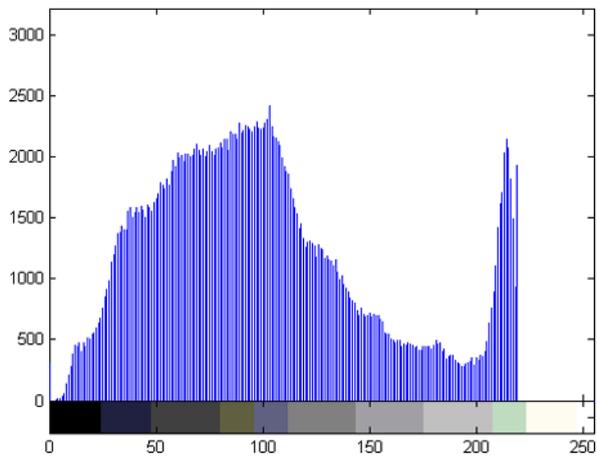
Fig. 7 The original image

all h_w in Class 2 and Class 3 by 2, the original difference values h are retrieved. Finally, the modified image I'_H via (12) is retrieved.

For each value of $k \in \{0, 1, 2, 3\}$, as long as the inequality (16) is satisfied, the h in Class 2 are classified into non-Flipping Class 2, otherwise, the h belong to Flipping Class 2. Bitstream \mathcal{H} is retrieved by extracting the LSBs of the remaining h_w in Class 3 and all h_w in non-Flipping Class 2, and then decompressed to retrieve the location map L_H . By identifying an end of message symbol, δ_1 and δ_2 are retrieved. Stage 2. With L_H , all histogram modified pixel values are found and listed orderly into a sequence, and then the sequence is divided into two intervals by 128. All the pixel values within the interval $[0, 127]$ are subtracted by the difference δ_1 , while all the values within the interval $[128, 255]$ are added by the difference δ_2 .



(a) The histogram image of 'Lena' image



(b) The histogram image of 'Goldhill' image

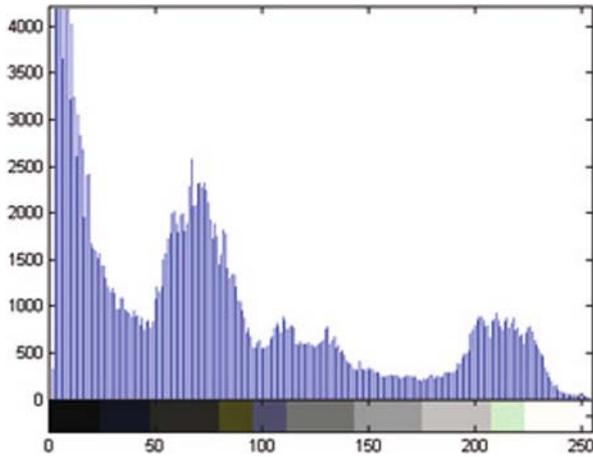
Fig. 8 The histogram image

After that, we perform IWT on the inverse histogram modified image. Data is extracted from all high-frequency wavelet coefficients according to (9). Assume the first bit in the extracted data is the companding errors r . The original C_h is retrieved according to (10).

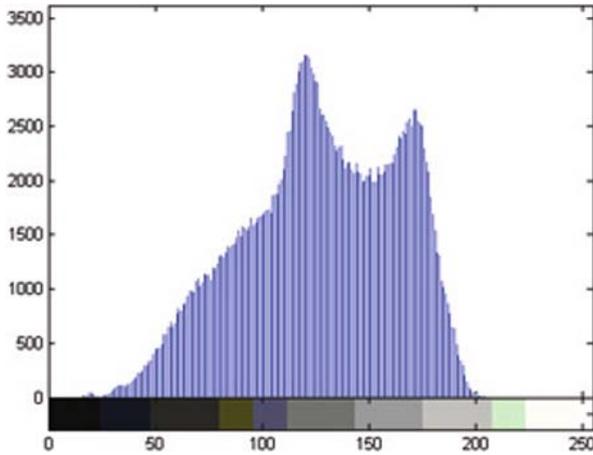
Finally, the inverse IWT is performed and the original image is retrieved.

5 Experimental Results

For the purpose of comparative study, we tested the proposed method and Xuan's method on several commonly used images, including 'Lena' of size 256×256 ,



(c) The histogram image of 'Build' image



(d) The histogram image of 'Baboon' image

Fig. 8 (Continued)

'Goldhill', 'Build' and 'Baboon' of size 512×512 . A binary random sequence derived from a uniform noise was used as a watermark signal in our experiments.

The PSNR values of the watermarked images are tabulated in Tables 2, 3, 4, and 5, along with the embeddable payload size. Figures 7 and 8 show the four original images and the histograms of pixel values, respectively.

The distribution of the gray-level value of pixels of an image, as reflected in its histogram, has interesting effects on the system performance. For the 'Lena' image and the 'Goldhill' image, there is no pixel distributed near the high end of the gray-level value spectrum. It can be seen that, from Table 2, the proposed method is far superior to Xuan's method in terms of both PSNR value and hiding capacity. With

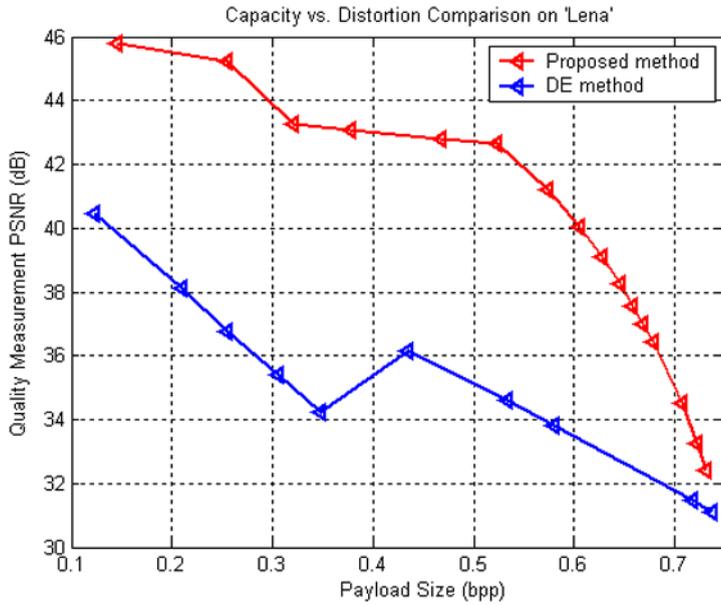


Fig. 9 Capacity versus distortion comparison of 'Lena'

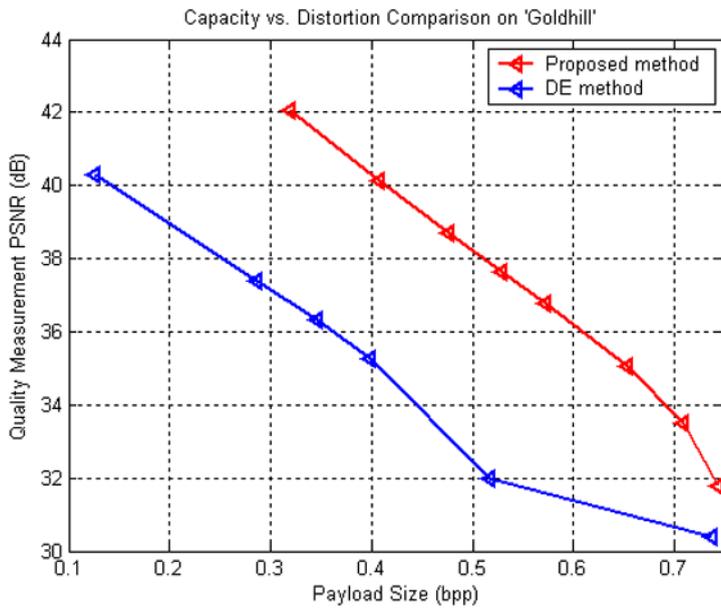


Fig. 10 Capacity versus distortion comparison of 'Goldhill'

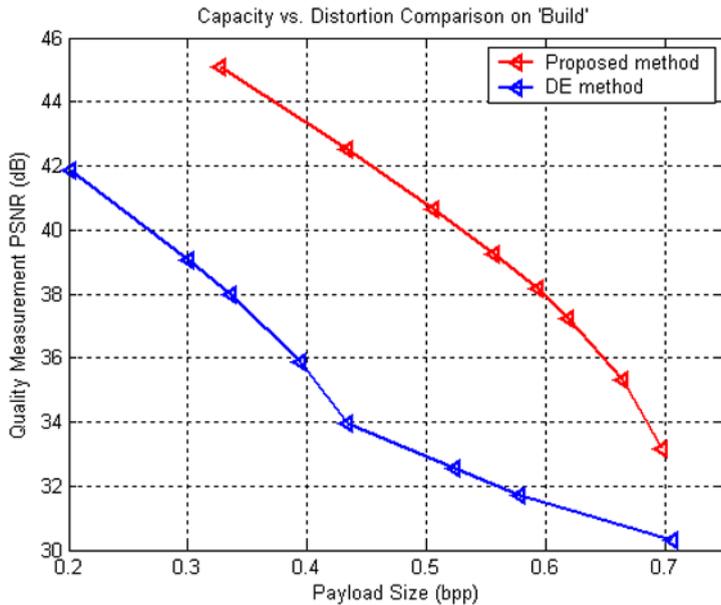


Fig. 11 Capacity versus distortion comparison of 'Build'

Xuan's method, the hiding capacity degrades when the threshold T_h increases. The reason is that the growth speed of the bitstream, produced by losslessly compressing the location map, further exceeds the number of high-frequency coefficients which are suitable for embedding. It is worthwhile to point out that the hiding capacity of the proposed method increases linearly as T_h increases.

There are images, 'Build' for instance, with pixels distributed across the entire gray-level spectrum. For images of this kind, the proposed method still performs better than Xuan's method. As reported in Table 4, the superiority is profound even with smaller T_h . 'Baboon' is a typical example of images for which both the high end and low end of the gray value spectrum are vacuum. According to the results reported in Table 5, for images of this sort, the performance of the proposed method is almost the same as that of Xuan's method.

Since we apply our improved DE method as a second embedding method, we implement the proposed method and the DE method in multiple embedding for comparative study. For multiple embedding in the DE method, the pairing direction was horizontal for the first embedding and vertical for the second embedding. The proposed method also can be applied to an image more than once for multiple data embedding. The capacity versus distortion comparisons for the proposed method and the DE method on 'Lena', 'Goldhill', 'Build', and 'Baboon' are shown in Figs. 9, 10, 11, and 12. It can be seen that the proposed method outperforms the DE method at an embedding rate around 0.75 bpp. The DE method outperforms ours beyond that.

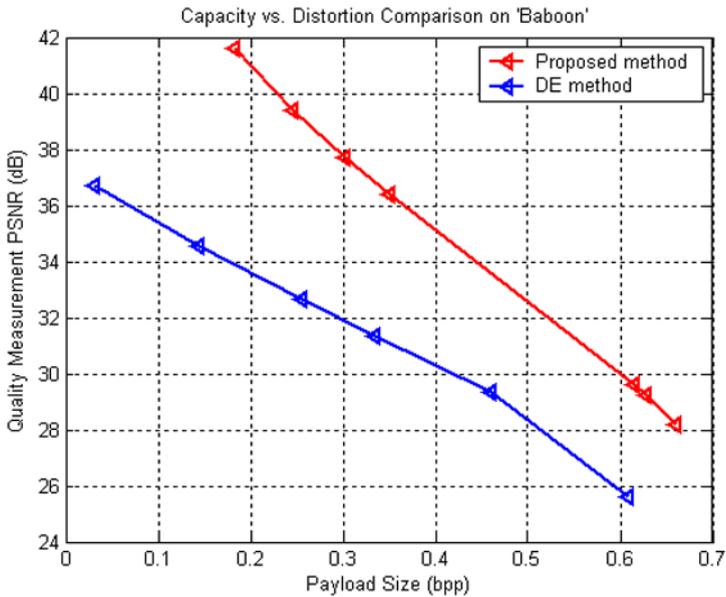


Fig. 12 Capacity versus distortion comparison of 'Baboon'

6 Conclusions

A reversible data hiding method based on the companding technique and an improved DE method is presented in this paper. By exchanging the order of wavelet transform and histogram modification, the hiding capacity can be considerably increased. Furthermore, an improved DE method is proposed for the embedding of additional information incurred by the histogram modification process. As revealed in the experimental results, the proposed scheme outperforms Xuan's method, in terms of embedding capacity and visual quality, as expected. An interesting phenomenon had been observed that the embedding capacity is strongly influenced by the gray-level histogram of an image. In the comparative study with the standard DE method, we conclude that the visual quality of marked images is better than that of the DE method for low to moderate embedding rates, while the DE method is better at high embedding rates.

References

1. A. M. Alattar, Reversible watermark using difference expansion of triplets, in *Proc. IEEE ICIP*, Barcelona, Spain, volume 1, 2003, page(s): 501–504.
2. M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, Reversible data hiding, in *Proc. IEEE ICIP*, volume 2, 2002, page(s): 157–160.
3. D. Coltuc, and J. M. Chassery, Simple reversible watermarking schemes: further results, in *Proc. SPIE, Security, Steganography, and Watermarking of Multimedia Contents*, 2006, page(s): 739–746.
4. D. Coltuc, and A. Tremeau, Simple reversible watermarking schemes: further results, in *Proc. SPIE, Security, Steganography, and Watermarking of Multimedia Contents*, 2005, page(s): 561–568.

5. J. Fridrich, M. Goljan, and R. Du, Invertible authentication, in *Proc. of SPIE, Security, Steganography and Watermarking of Multimedia Contents*, San Jose, volume 3971, 2001, page(s): 197–208.
6. M. Goljan, J. Fridrich, and R. Du, Distortion-free data embedding for images, in *4th Information Hiding Workshop*, Springer, New York, LNCS, volume 2137, 2001, pages(s): 27–41.
7. C. W. Honsinger, P. Jones, M. Rabbani, and J. C. Stoffel, *Lossless Recovery of an Original Image Containing Embedded Data*, US patent, No. 77102/E-D, 1999.
8. B. Macq, Lossless multiresolution transform for image authenticating watermarking, in *Proc. EU-SIPCO*, 2000, page(s): 533–536.
9. J. Tian, Reversible data embedding using a difference expansion, *IEEE Trans. Circuits Syst. Video Technol.*, volume 13, no. 8, 2003, page(s): 890–896.
10. M. Veen, F. Bruekers, A. Leest, and S. Cavin, High capacity reversible watermarking for audio, in *Proc. SPIE, Security, Steganography, and Watermarking of Multimedia Contents*, volume 5020, 2003, page(s): 1–11.
11. G. Xuan, C. Yang, Y. Zhen, Y. Shi, and Z. Ni, Reversible data hiding using integer wavelet transform and companding technique, *IWDW*, 2004.
12. B. Yang, M. Schmucker, W. Funk, C. Brush, and S. Sun, Integer DCT-based reversible watermarking for images using companding technique, in *Proc. of SPIE, Security, Steganography and Watermarking of Multimedia Contents*, San Jose, USA, 2004, page(s): 405–415.