Cheng et al. EURASIP Journal on Image and Video Processing (2018) 2018:74
https://doi.org/10.1186/s13640-018-0306-6

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

An efficient histogram-preserving steganography based on block

Cheng Jie, Chen Zhenzuo and Yang Rener*

Abstract

In order to reduce the influence on cover image caused by embedding algorithm, while keeping the histogram characteristics of cover image, in this paper a histogram preserving steganography algorithm is proposed based on dividing a secret message into blocks. The secret message and cover image are divided into blocks and the hamming distance of least significant bits is calculated, and then secret message block and mark-bit of cover image block are processed according to the hamming distance. In terms of histogram profit-lost compensation, the processed secret message block is then embedded into the cover image block. Theoretical derivation of modification gain and embedding efficiency in the algorithm is well verified by tests. Compared with the matrix coding steganography algorithm, which also has less unwanted effect on the cover image, the steganography algorithm proposed in this paper achieves higher embedding efficiency, higher capacity, and lower computational complexity other than high histogram consistency. A tradeoff control between embedding capacity and image distortion can be flexibly manipulated at nearly no or little cost of compromising the histogram characteristics.

Keywords: Steganography, Histogram preserving, Embedding efficiency, Computational complexity, Histogram distortion, Embedding rate

1 Introduction

With the development of modern society, the information security problems identified with various communication means are attention-drawing in the research communities. As an important branch of information security domains, steganography [1–4] takes advantage of redundancy in digital media for information-hiding to better secure secret information transmission.

To mainly improve the embedding capacity and efficiency, the matrix embedding was introduced by Ron [5]. The linear coding is a type of steganography scheme based on matrix embedding [5–7], which explores the linear relation between the cover image matrix and the secret information matrix to implement the concealment of secret information. Due to its embedding efficiency and easiness for implementation, the matrix embedding is adopted by mainstream steganography algorithms such as F5 [7] and edge adaptive image steganography based on least significant bit matching revisited (EALSBMR) [8]. The good performance by steganography algorithm reduces the alterations of the cover image, including the distortion, to its minimal level when embedding the secret information of the same size. Ron [5] first proposed a steganography scheme based on binary hamming code. In this scheme, more than one binary bit of the secret information can be embedded by modifying only one binary bit of the cover image, and the advantage is noted for high embedding efficiency and large embedding capacity. However, the merits come along with the side effect of significant distortion. To tackle this issue, on the basis of matrix embedding algorithm, in Ref. [2, 6, 9–25], the researchers attempted to improve the coding scheme aiming at reducing the additive embedding distortion and computational complexity. In Ref. [9], Fridrich et al. proposed the wet paper coding scheme based on linear matrix embedding, which selects those bits of cover image to be modified according to some specific rules. To a certain extent, this coding scheme reduces the distortion degree. However, the embedding efficiency is compromised as a result. In the literature [13], the Reed-Muller encoding is used for

* Correspondence: jackcheng18920@163.com
College of Information Science and Engineering, Ningbo University, Zhejiang, China

matrix embedding in the binary image, and the higher embedding efficiency is achieved. When using [13] to execute the matrix embedding algorithm, selecting the suitable embedding position on the cover image is also considered a way to reduce distortion. Literature [26] applies the graph matching theory to determine the modification position and modification mode of the cover image. In literature [27], edge detection is used to extract the boundary pixels in the cover image, and the bits of which are tagged as modifiable ones. Compared with Ron's hamming code embedding, the above two methods can well dampen the distortion of image, but the performance of the algorithm is sensitive to the properties of cover image, which may lead to a sharp decline in embedding capacity and a notable increase in algorithm complexity. In [11], parity check matrix using hamming code is rearranged among the columns in the matrix to lower the computational complexity to the satisfying degree. However, the embedding rate is fixed and the embedding capacity is not as high as expected.

On the other hand, in order to reduce the data embedding distortion, some steganographic schemes capable of preserving original data characteristics have been also developed [28–32]. Using the techniques described in [31, 32], the secret message can be embedded into least significant bits of cover image when keeping the original histogram distribution intact. In [30], the model-based steganography ensures that the distribution model of stego data is consistent with that of cover data. Furthermore, the maximization of payload with a low distortion level is taken into account in Ref. [28, 29]. On the premise of the correlation between the flip probability and the original value, the payload-distortion performance approaches the theoretical limit in [29]. At the cost of high computational complexity, steganographic scheme in [28] maximizes the payload without compromising the histogram of original data and image fidelity.

Inspired by these afore-mentioned works, we propose a type of steganography algorithm with high efficiency and low complexity. In order to keep the statistical features of images, we exploit histogram-preserving steganography (HPS) [33], in the process of steganography. The remainder of this paper is organized as follows: Section 2 briefly introduces related works, such as matrix embedding, matrix extending method, revised hamming code embedding, and histogram-preserving embedding; the proposed histogram-preserving steganography based on block (HPSB) is described in Section 3 in detail. Theoretic derivations are presented in Section 4. Experimental results and discussion are covered in Section 5. Conclusion is made and some prospective works are suggested in Section 6.

2 Related work

2.1 Matrix embedding

Let $\mathbf{H}[n-k, n]$ denote parity check matrix (PCM) of a binary linear $[n, k]$ code C of length n, where k is the length of one dimension binary vector [5, 7]. Use $c^T = (c_1, c_2, \cdots, c_n) \in F^n(2)$ to denote cover bits, where $F^n(2)$ represents the Galois field of order 2 and length n, i.e., a space of all n-bit column vectors, e.g., $x = (x, x, \cdots, x_n)$, where c^T means transpose of matrix $\boldsymbol{c}$. In the same way, use $m^T = (m_1, m_2, \cdots, m_{n-k}) \in F^{n-k}(2)$ to denote secret bits. As assumed above, the matrix embedding is performed using the following means. Firstly, calculate the difference between $\mathbf{Hc}$ and $\boldsymbol{m}$ with exclusive-or operation, i.e., $u \triangleq^{\text{def}} \mathrm{H}c \oplus m$, namely syndrome. Secondly, solve the system of linear eq. $\mathrm{H}x = \mathrm{H}c \oplus m$ to find a solution vector $x_{\min}$, namely coset leader, such that $x_{\min} = \arg\min_{\{x \in F^n(2) | \mathrm{H}x = \mathrm{u}\}} \omega(x)$, where $\omega(x)$ is hamming weight of vector x. The final stego bits of $\boldsymbol{s}$ can be obtained using $s = c \oplus x_{\min}$. The secret message $\boldsymbol{m}$ can be extracted correctly by using the formula (1).

$$\begin{aligned} \mathrm{H}s &= \mathrm{H}(\mathrm{c} \oplus x_{\min}) = \mathrm{Hc} \oplus \mathrm{H}x_{\min} = \mathrm{Hc} \oplus u \\ &= \mathrm{Hc} \oplus (\mathrm{Hc} \oplus m) = m \end{aligned} \tag{1}$$

Since the algorithm must solve a set of $n-k$ linear equations with n unknowns in $F^n(2)$, thus the computational complexity is very high.

2.2 Fast algorithm using hamming code

Because the computational complexity of matrix embedding is $O(n2^k)$, it is a real nondeterministic polynomial problem when n and k is considerably large. In Ref. [11], Mao pointed out that for the random linear code [9], since there is an identity matrix of $n-k \times n-k$ in the left part of its parity check matrix $\mathbf{H}$, the form of which is the same as denoted in Section 2.1, some coset leaders in the form of $[u^{\mathrm{T}}, z]$ can be identified by the syndromes themselves, where u is syndrome as mentioned in Section 2.1 and z is $1 \times k$ vector, the elements of which are all zeroes. So, the computational complexity of embedding algorithm using random linear code can be mitigated to a certain degree.

For hamming code part, Mao also took advantage of two criterions to simplify matrix embedding. Here are the criterions:

(1) The syndromes and their coset leaders are one to one correspondent.
(2) Changing the positions of any two columns of the parity check matrix $\mathbf{H}$ does not change the characteristics of the Hamming code.

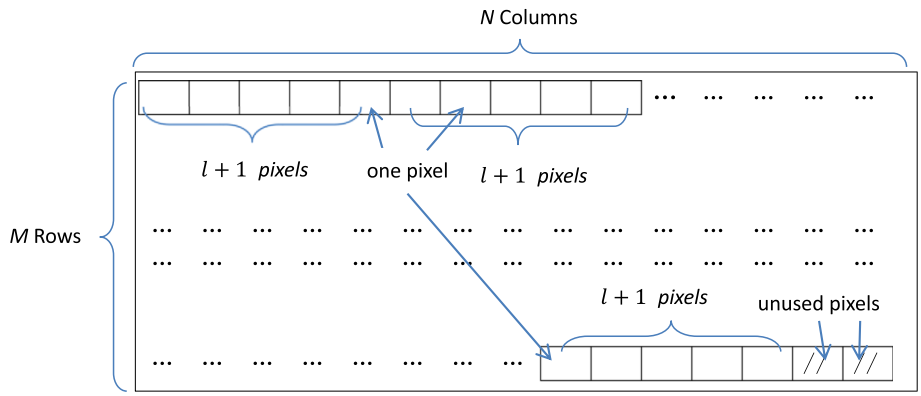

Fig. 1 Decompose cover image

On account of ergodic property of the columns of the parity check matrix **H** of a hamming code, which takes all the possible permutations of 0 and 1 (except all zeros), Mao rearranged H by the decimal forms of matrix **H**'s column running from 1 to $2^{n-k}-1$. For instance, rearranged parity check matrix H of $[7, 4]$ hamming code is in the following form:

$$\mathrm{H} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

For example, assume that the cover bits is vector $c = [1000000]$ and the secret data is $m = [100]$, then $\mathrm{H}c \oplus m = [101]$. Since $[101]$ is 5 in the decimal form, the fifth bit in $\boldsymbol{c}$ should be changed. Therefore, the stego bits are vector $s = [1000100]$. At the receiver side, the extracted vector is $\mathrm{H}s = [100]$, that is the conveyed secret data $\boldsymbol{m}$ exactly.

In so doing, the syndromes themselves indicate the coset leaders; therefore, the computational complexity of Mao's algorithm is reduced to O(1) perfectly. Other than the obvious advantage, in terms of the design nature, the embedding rate is fixed and the embedding capacity is not as high as expected.

2.3 Histogram-preserving algorithm

In [28], Zhang proposed a method of efficient data-embedding in binary sequence. The steps of embedding secret k bits into cover $k+16$ bits are as follows:

(1) Divide the secret sequence into a series of bit blocks, each of which contains k bits, e.g., $s = [s_1, s_1, \cdots, s_k]$;
(2) Generate a matrix G of $[\mathbf{Q}, \mathbf{I}_{16}]$, where Q is a $16 \times k$ pseudo-random binary matrix and $\mathbf{I}_{16}$ is a 16×16 identity matrix;
(3) According to the following formula:

$$\mathrm{v}_i = [s, 0_{16}] + b_i \cdot G \tag{2}$$

where $\boldsymbol{s}$ is secret k bits, 0_{16} is zero array of length 16, and $\mathrm{b}_i \in \mathrm{F}^{16}(2)$, $i = 0, 1, \cdots, 2^{16}-1$, the algorithm produces 2^{16} different types of binary candidate vectors, the length of which are $k+16$. Denote matrix **H** as $[\mathbf{I}_k, \mathbf{Q}^T]$, where $\mathbf{I}_k$ is a $k \times k$ identity matrix and T-operation is matrix transpose. So, $\boldsymbol{s}$ can be derived from formula of $s = \mathrm{H} \cdot \mathrm{v}_i^T$;

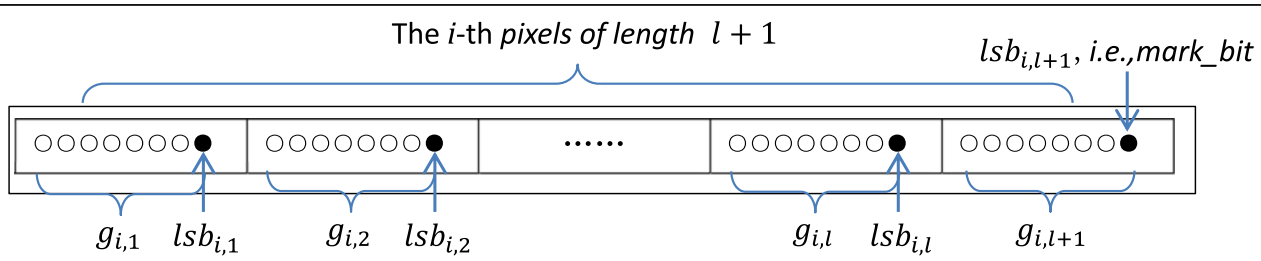

Fig. 2 Structure of $l+1$ pixels

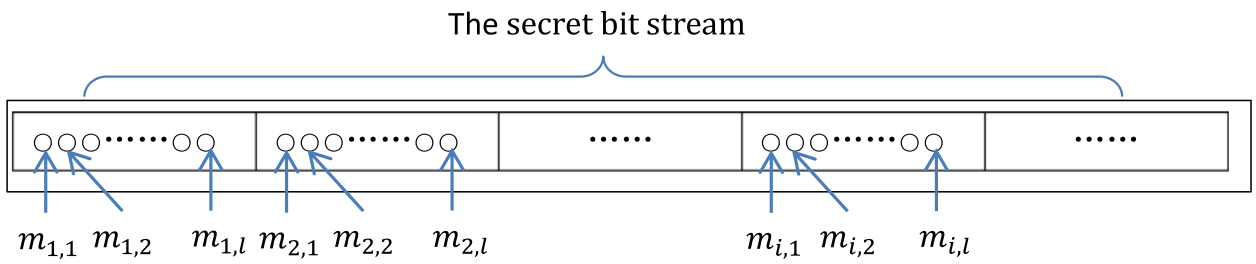

Fig. 3 Structure of the secret bit stream

(4) For each candidate vector v_i, in accordance with presetting probabilities of 0s and 1s, convert v_i *to* v_i'. Denote α probability of 0 and $1-\alpha$ that of 1 so as to ensure the bit distribution in the candidate stego vectors v_i' to be similar to that in the original binary cover sequence;
(5) Solve the optimal problem as follows:

$$\mathrm{r}_{\min} = \arg \min_{i\in[0,\ 2^{16}-1]} n_{i,1} + n_{i,0} + \lambda\cdot(S_1 - S_0)\cdot(n_{i,1} - n_{i,0}) \quad (3)$$

Here, $n_{i,\ 1}$ is the number of positions where the cover bit is 1 and the bit in v_i' is 0, and $n_{i,\ 1}$ is for the opposite situation. The bits in v_i' and the corresponding bits in the cover sequence are compared in a bit-by-bit manner. Denote the number of cover bits that have been flipped from 1 to 0 as S_1, and the opposite is marked as S_0. Their initial values are both zero;

(6) Replace the $k+16$ bits in cover sequence with v'_{opt} that has $r_{\min}$. In the meantime, update S_1 and S_0.

By using opposite operation, data extraction is easier than embedding. In fact, Zhang's algorithm commendably maintains the statistical characteristics of histogram of stego image. The algorithm well controls the distortion level while preserving the original distribution of each cover sequence, while obtains high payload. Nevertheless, when embedding secret k bits into cover $k+16$ bits at every turn, it is substantially time-consuming to find out the most optimal vector v'_{opt} that satisfies $r_{\min}$ among 2^{16} candidate vectors, along with executing $k+16$ calculations in order to convert v_i to v_i' for each v_i.

3 Proposed method

This section introduces histogram-preserving steganography based on block (HPSB). HPSB algorithm implementation process: firstly, both secret message and cover image are divided into blocks; then by considering hamming distance of least significant bit, the secret message and cover image are preprocessed respectively; next, in terms of histogram profit-lost compensation, the preprocessed secret message is embedded into the preprocessed cover image. Before applying HPSB algorithm, the tracker of histogram profit-lost compensation should be initialized. While embedding each bit, the tracker makes embedding decision, and the value of the tracker is constantly updated simultaneously. Block by block, the secret message is embedded into cover image until steganographic procedure is completed (Figs. 1, 2, 3, 4, 5 and 6). HPSB algorithm is presented in detail as follows:

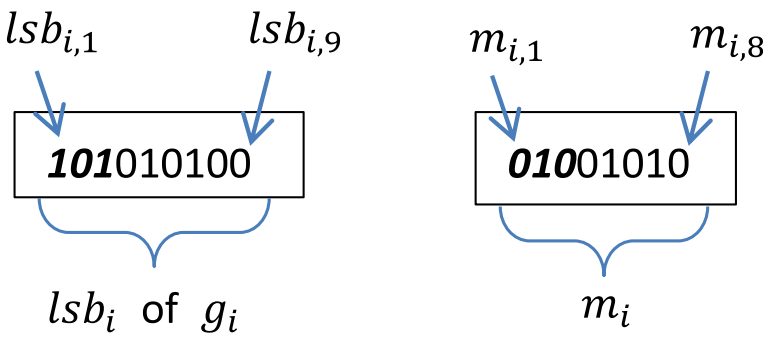

Fig. 4 Illustration for $d_i \le l/2$. The length l is 8 and d_i is 3. So embedding bits is 01001010 and $lsb_{i,9}$ will be set to 0

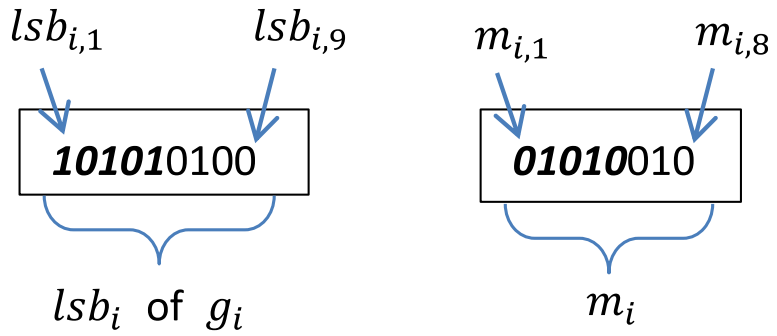

Fig. 5 Illustration for $d_i > l/2$. The length l is 8 and d_i is 5. So embedding bits is 10101101 and $lsb_{i,9}$ will be set to 1

(1) Decompose L-pixels cover image into $[L/(l+1)]$ blocks, the length of which is $l+1$, where [.] denotes the floor operator. Use g_i $(i = 1, 2, \ldots, [L/(l+1)])$ to denote blocks, and pixel value in a block can be denoted as $g_{i,1}, g_{i,2}, \ldots, g_{i,l}, g_{i,l+1}$. Least significant bit of each pixel can be denoted as $lsb_{i,1}, lsb_{i,2}, \ldots, lsb_{i,l}, lsb_{i,l+1}$, and specially $lsb_{i,l+1}$ is the mark-bit for the block of g_i.

(2) Compare least significant bits of g_i, namely, $lsb_{i,1}, lsb_{i,2}, \ldots, lsb_{i,l}, lsb_{i,l+1}$, with a corresponding binary secret message, namely, $m_{i,1}, m_{i,2}, \ldots, m_{i,l}$, and calculate hamming distance d_i, using formula (4) and (5) defined as follows:

$$d_i = l - \sum_{j=1}^{l} \theta(lsb_{i,j}, m_{i,j}) \tag{4}$$

$$\theta(lsb_{i,j}, m_{i,j}) = \begin{cases} 0 & if\ lsb_{i,j} \neq m_{i,j} \\ 1 & if\ lsb_{i,j} = m_{i,j} \end{cases} \tag{5}$$

where d_i denotes the count that the number of least significant bits of g_i are different from corresponding bits of m_i .

(3) According to hamming distance d_i as mentioned, condition check and embedding operation are performed as follows:

(3-1) While $d_i \le l/2$, set mark-bit of $lsb_{i,l+1}$ to 0, and employ HPS method to conceal $(m_{i,1}, m_{i,2}, \ldots, m_{i,l})$ into $(g_{i,1}, g_{i,2}, \ldots, g_{i,l})$. Thus the least significant bits of g_i become $(m_{i,1}, m_{i,2}, \ldots, m_{i,l}, 0)$.

(3-2) While $d_i > l/2$, set mark-bit of $lsb_{i,l+1}$ to 1, and employ HPS method to conceal $(\overline{m_{i,1}}, \overline{m_{i,2}}, \ldots, \overline{m_{i,l}})$ into $(g_{i,1}, g_{i,2}, \ldots, g_{i,l})$, where $\overline{m_{i,1}}$. is non-operation on $m_{i,1}$. Thus the least significant bits of g_i become $(\overline{m_{i,1}}, \overline{m_{i,2}}, \ldots, \overline{m_{i,l}}, 1)$.

In detail, the above embedding algorithm consists of two sub-functions as follows:

(3-a) For $(m_{i,1}, m_{i,2}, \ldots, m_{i,l})$ and $(lsb_{i,1}, lsb_{i,2}, \ldots, lsb_{i,l})$, calculate $\delta(lsb_{i,j}, m_{i,j})$ using the following formula (6).

$$\delta(lsb_{i,j}, m_{i,j}) = \begin{cases} 0 & lsb_{i,j} = m_{i,j} \\ \pm 1 & lsb_{i,j} \neq m_{i,j} \end{cases} \tag{6}$$

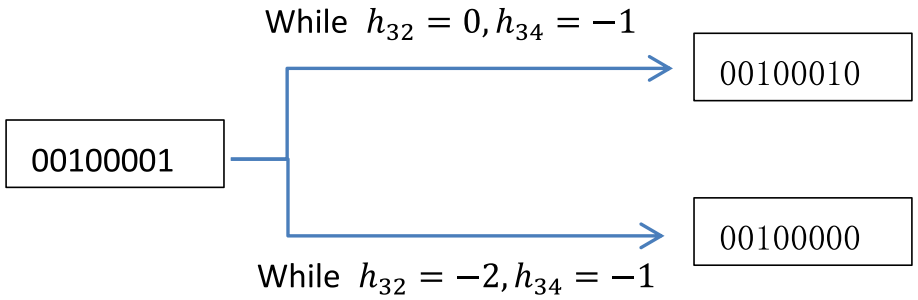

Fig. 6 Assume $g_{i,j}$ is 00100001. Embed 0 into $g_{i,j}$

(3-b) Steganography function is $f(g_{i,j}, m_{i,j}) = g_{i,j} + \delta(lsb_{i,j}, m_{i,j})$, when $d_i \le l/2$ is satisfied, otherwise is $f(g_{i,j}, m_{i,j}) = g_{i,j} + \delta(lsb_{i,j}, \bar{m}_{i,j})$, where $f(g_{i,j}, m_{i,j})$ represents stego pixel value, and $\delta(lsb_{i,j}, m_{i,j})$ represents indicator of profit-lost compensation.

(4) Histogram-preserving algorithm exploits statistics of pixels value to add one to $\delta(lsb_{i,j}, m_{i,j})$ or conversely subtract one from $\delta(lsb_{i,j}, m_{i,j})$. Strategy of histogram-preserving algorithm is to minimize the histogram discrepancy between cover image and stego image. Use histogram-preserving strategy to determine the $+1$ or -1 operation, so that histogram deviation caused by previous embedding process, i.e., the number of pixels whose value is decreased is much more than that is increased, can be

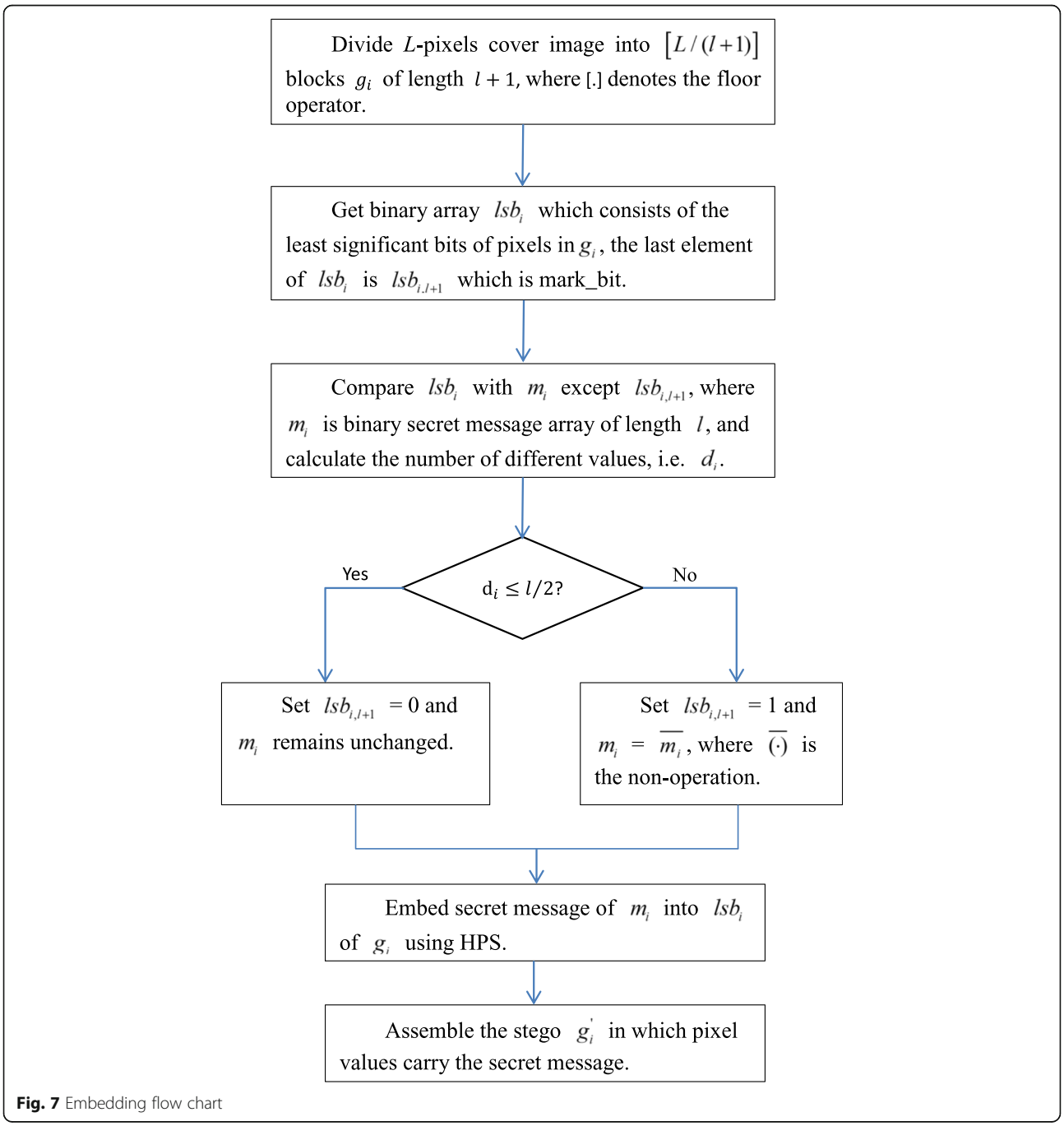

Fig. 7 Embedding flow chart

compensated in time. At the beginning of embedding, initialize $h_k = 0$, $k = 0, 1, 2, \ldots, 255$, where h_k denotes the number of pixel value of k at one point of embedding procedure. Moreover, h_k serves as the basis for the different assignment of $+1$ or -1 on $\delta(lsb_{i,j}, m_{i,j})$ when $lsb_{i,j} \neq m_{i,j}$ is satisfied.

The value of h_k indicates the increase or decrease in the number of pixels with a value of k. For example, if h_{33} is 2 or -1, this means that after embedding the secret message the number of pixels with the value of 33 is increased by 2 or decreased by 1 compared to the original cover image. The closer the value of h_k is to 0, the less the histogram changes. Assuming the value of pixel $g_{i,j}$ is 33, when $lsb_{i,j} = m_{i,j}$, the value of h_{33} remains unchanged, and when $lsb_{i,j} \neq m_{i,j}$, the value of h_{33} is decrease by 1 since the value of pixel $g_{i,j}$ is to be altered. If the value of pixel $g_{i,j}$ is either decreased to 32 or increased to 34, the value of either h_{32} or h_{34} is increased by 1. Either $+1$ or -1 to $g_{i,j}$ depends on the following strategy: (1) If the two elements adjacent to h_k have opposite signs, the value of $g_{i,j}$ is altered towards the negative element. For instance, if h_{32} is -2 and h_{34} is 1, $g_{i,j}$ is altered to 32 and h_{32} is increased by 1. (2) If the two elements adjacent to h_k have the same signs, the value of $g_{i,j}$ is altered towards the smaller element. For instance, if h_{32} is 2 and h_{34} is 1, $g_{i,j}$ is altered to 34 and h_{34} is increased by 1. (3) If the two elements adjacent to h_k have the same value, the modification direction is random

The embedding flow is shown in Fig. 7.

The extraction operation is an inverse process and is convenient to perform. Decompose the stego data into blocks and pick out the mark-bit to determine whether applying non-operation on least significant bits or not. The extraction flow is shown in Fig. 8. In this way, receiver can extract secret data from stego data. In addition, because of histogram distribution preserved, the secret data is more secure and the stego data has better immunity to attack of steganalysis.

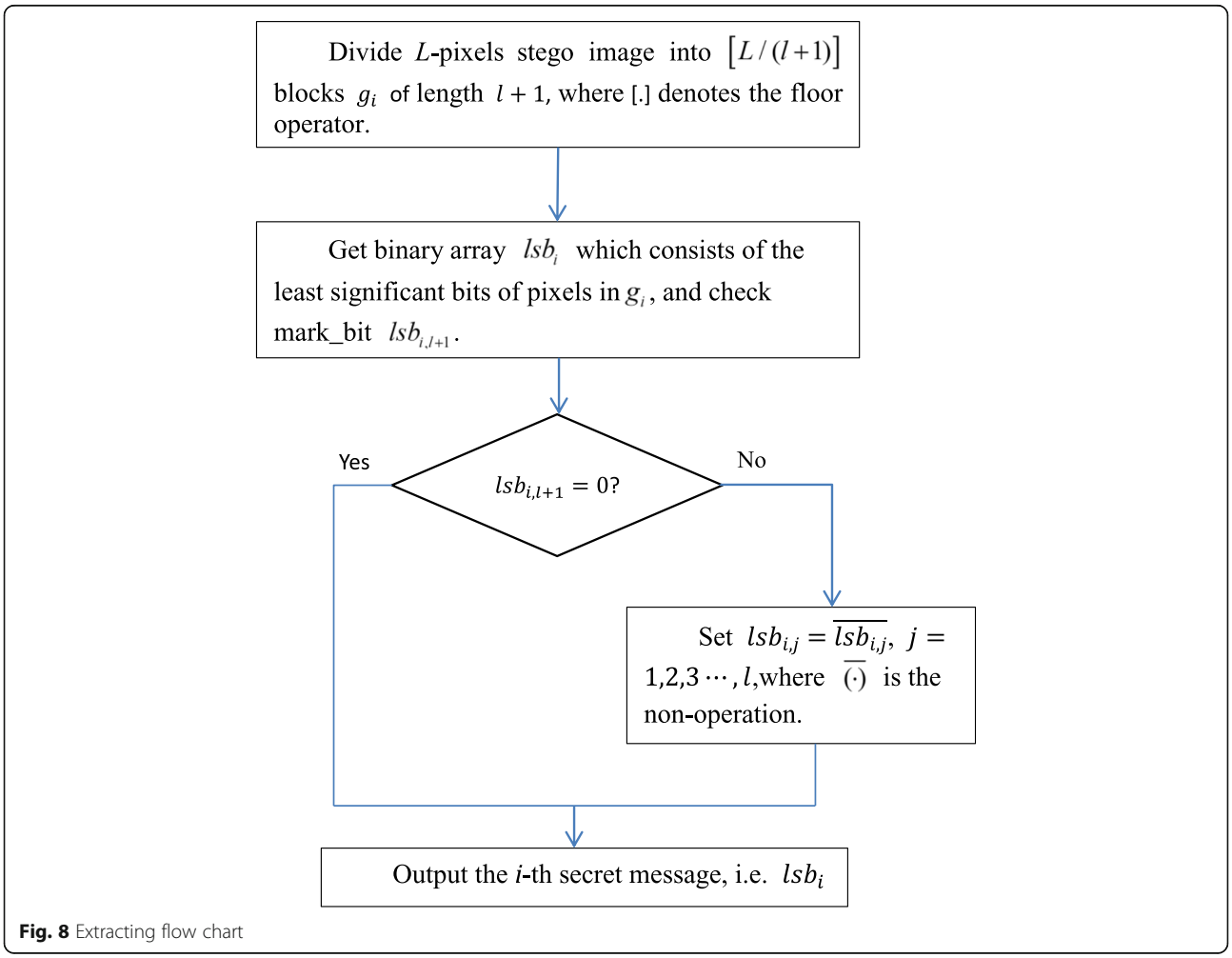

Fig. 8 Extracting flow chart

4 Theoretical analysis of HPSB steganography algorithm

Embedding rate (ER) is defined as $\mathrm{ER} = \frac{k}{n}$, when k bits secret information are embedded into n bits cover data along with d bits cover data modified. Furthermore, distortion rate (DR) and embedding efficiency (EE) are respectively defined as $\mathrm{DR} = \frac{d}{n}$ and $\mathrm{EE} = \frac{k}{d}$. ER denotes the ratio of the quantity of embedded secret data to that of cover data. DR denotes the ratio of the quantity of modified bits to that of cover bits. The higher the embedding rate, the more secret information is embedded. However, as embedding rate goes up, visual imperceptible and anti-statistical detection performance is compromised. In order to improve the security, it is worthy of consideration that the embedding efficiency should be as high as possible at a given embedding rate. Obviously, in accordance with the presented method in Section 3, the number of modification of cover image pixels is reduced and the embedding efficiency is improved. As a result, at nearly no or little cost of reducing embedding capacity, the average distortion is also decreased.

The rest of this section is the analysis of the proposed algorithm. In general, suppose that set of pseudo random binary bit sequence is embedded into cover image using the least significant bit (LSB) method, since the number of difference between bits in the secret sequence and corresponding least significant bits in the image pixel is approximately equal to 50% of total, the probability of LSB modification is nearly half. In other words, the number of equivalence between bits in the secret sequence and corresponding least significant bits in the image pixel obeys binomial distribution $B(l, 0.5)$, where l is the length of block. Thus, for the same length l of bits, the modification expectancy of LSB-based algorithm is formula (7)

$$\mathrm{md} = \sum_{i=0}^{l} \mathrm{C}_l^i \left(\frac{1}{2}\right)^l \cdot i \tag{7}$$

and that of HPSB algorithm is formula (8)

Table 1 Performance of HPSB algorithm with different block length l

l	PSNR (dB)	ER	DR	EE
2	52.93	0.667	0.331	2.0162
3	53.18	0.750	0.313	2.4000
4	52.70	0.800	0.349	2.2931
5	52.77	0.833	0.343	2.4239
6	52.50	0.857	0.365	2.3462
7	52.53	0.875	0.363	2.4088
8	52.36	0.889	0.378	2.3515
9	52.37	0.900	0.377	2.3877
10	52.25	0.909	0.388	2.3444
11	52.25	0.917	0.387	2.3677
12	52.16	0.923	0.396	2.3334
13	52.16	0.929	0.395	2.3496
14	52.09	0.933	0.402	2.3213
15	52.09	0.938	0.402	2.3335

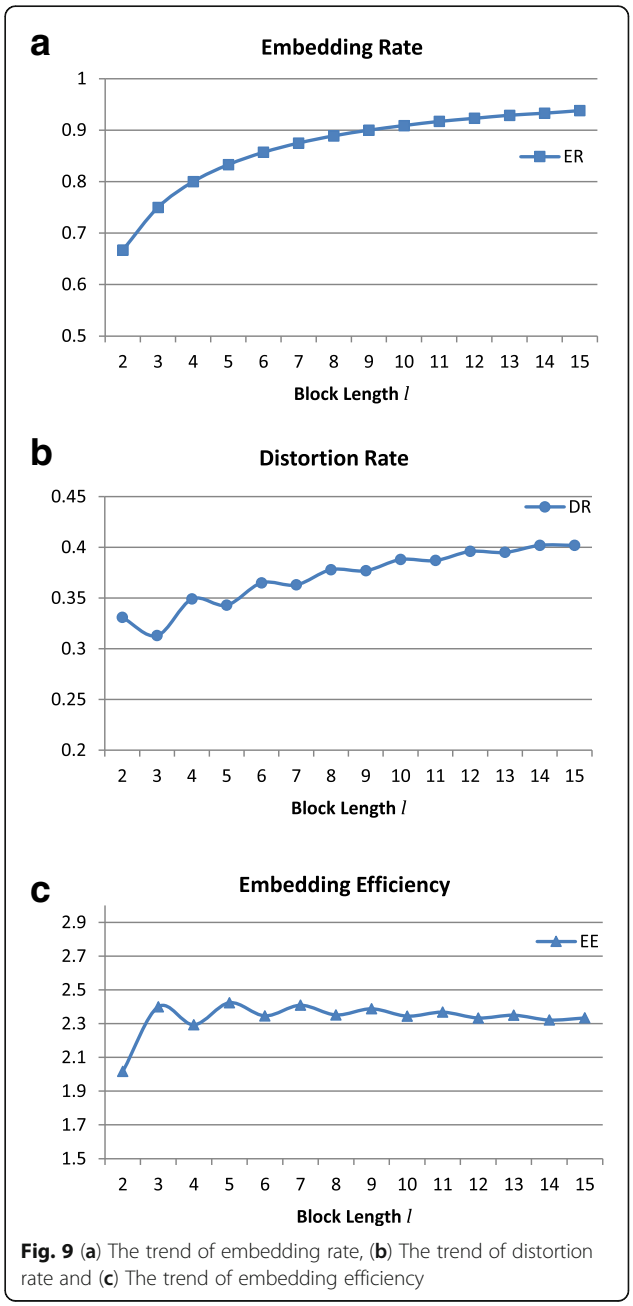

Fig. 9 (**a**) The trend of embedding rate, (**b**) The trend of distortion rate and (**c**) The trend of embedding efficiency

Table 2 Comparison of experimental and theoretical value of MGR and EE

l	MGR actual	MGR($\frac{C_{2m}^{m}}{2^{2m}}$) theoretical	MGR($\frac{1}{\sqrt{\pi m}}$) approximate	EE actual	EE theoretical
2	0.5001	0.5000	0.5642	2.0162	2.0000
3	0.5000	0.5000	0.5642	2.4000	2.4000
4	0.3751	0.3750	0.3989	2.2931	2.2857
5	0.3751	0.3750	0.3989	2.4239	2.4242
6	0.3127	0.3125	0.3257	2.3462	2.3415
7	0.3126	0.3125	0.3257	2.4088	2.4086
8	0.2735	0.2734	0.2821	2.3515	2.3486
9	0.2735	0.2734	0.2821	2.3877	2.3876
10	0.2461	0.2461	0.2523	2.3444	2.3422
11	0.2462	0.2461	0.2523	2.3677	2.3674
12	0.2258	0.2256	0.2303	2.3334	2.3317
13	0.2257	0.2256	0.2303	2.3496	2.3492
14	0.2094	0.2095	0.2132	2.3213	2.3203
15	0.2096	0.2095	0.2132	2.3335	2.3332

$$\mathrm{md}' = \sum_{i=0}^{[l/2]} C_l^i \left(\frac{1}{2}\right)^l \cdot i + \sum_{i=[l/2]+1}^{l} C_l^i \left(\frac{1}{2}\right)^l \cdot (l-i) \tag{8}$$

where $[\cdot]$ denotes floor operator. Then the excepted decrement of bit modifications, i.e., $E(\Delta d) \triangleq \mathrm{md} - \mathrm{md}'$, is depicted in detail as follows:

$$\begin{aligned} E(\Delta d) &= \left(\frac{1}{2}\right)^l \cdot \sum_{i=[l/2]+1}^{l} C_l^i (2i-l) \\ &= \left(\frac{1}{2}\right)^l \cdot \left(2 \cdot \sum_{i=[l/2]+1}^{l} C_l^i \cdot i - \sum_{i=[l/2]+1}^{l} C_l^i \cdot l\right) \\ &= \left(\frac{1}{2}\right)^l \cdot 2 \cdot \sum_{i=[l/2]+1}^{l} \frac{l! \cdot i}{i!(l-1)!} - \left(\frac{1}{2}\right)^l \cdot l \cdot \sum_{i=[l/2]+1}^{l} C_l^i \\ &= \left(\frac{1}{2}\right)^l \cdot 2l \cdot \sum_{k=[l/2]}^{l-1} C_{l-1}^k - \left(\frac{1}{2}\right)^l \cdot l \cdot \sum_{i=[l/2]+1}^{l} C_l^i \end{aligned} \tag{9}$$

(a) When l is an odd number, i.e., $l = 2m + 1$, the binomial expansion can be calculated:

$$\sum_{i=m+1}^{l} C_l^i = 2^{l-1} \tag{10}$$

$$\sum_{k=m}^{l-1} C_{l-1}^k = \frac{\left(2^{l-1} + C_{l-1}^m\right)}{2} \tag{11}$$

then formula (9) can be simplified to formula (12) as follows:

$$E(\Delta d) = \frac{C_{2m}^m}{2^{2m}} \cdot \frac{1}{2} l \tag{12}$$

(b) When l is an even number, i.e., $l = 2m$, the binomial expansion can be calculated:

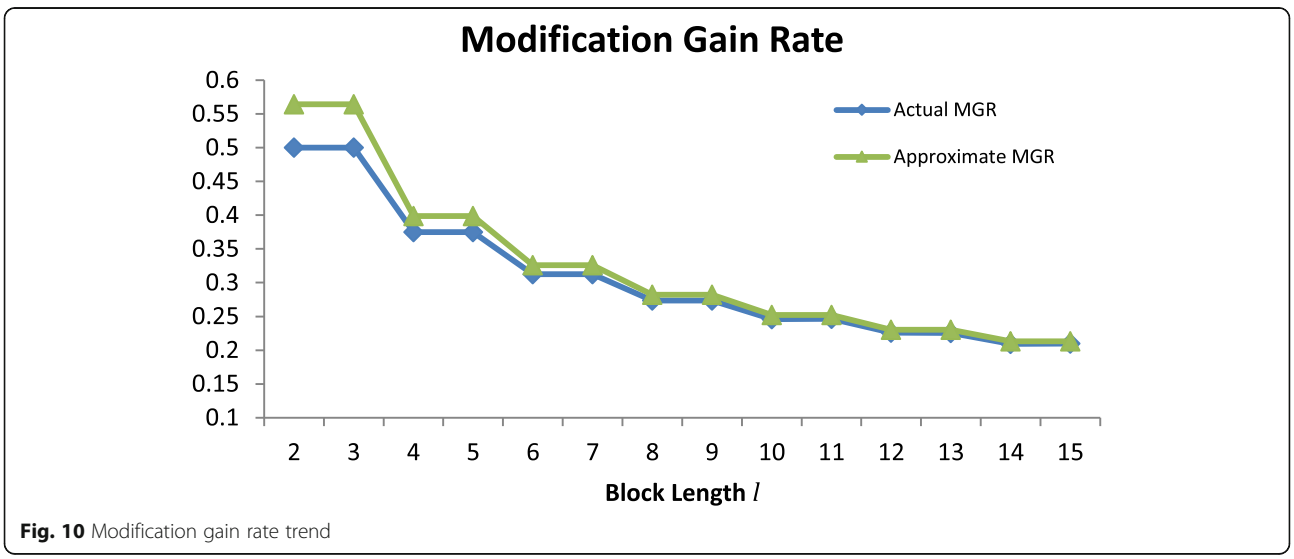

Fig. 10 Modification gain rate trend

$$\sum_{k=m}^{l-1} C_{l-1}^{k} = 2^{l-1}/2 \quad (13)$$

$$\sum_{k=m+1}^{l} C_{l}^{i} = 2^{l-1} - \frac{1}{2} C_{l}^{m} \quad (14)$$

then formula (9) can be simplified to formula (15) as follows:

$$E(\Delta d) = \frac{C_{2m}^{m}}{2^{2m}} \cdot \frac{1}{2} l \quad (15)$$

So for any l value,

$$E(\Delta d) = \frac{C_{2m}^{m}}{2^{2m}} \cdot \frac{1}{2} l \quad (16)$$

where $m = \left[\frac{l}{2}\right]$ and $[\cdot]$ is floor operator. When l is large, use Stirling formula (17) to simplify computation.

$$m! = \sqrt{2\pi m}\left(\frac{m}{e}\right)^{m} \quad (17)$$

Thus, formula (16) can be further simplified to formula (18) as follows:

$$E(\Delta d) = \frac{1}{\sqrt{\pi m}} \cdot \frac{1}{2} l \quad (18)$$

To conceal the secret information of length l, when using the LSB embedding method, the average modification quantity of the pixel value is $\frac{1}{2}l$, and as a comparison, when using embedding scheme proposed in this paper, the quantity of modification is reduced by $\frac{C_{2m}^{m}}{2^{2m}} \cdot \frac{1}{2} l$. Therefore, modification gain rate (MGR) is described as follows:

$$\mathrm{MGR} = \frac{C_{2m}^{m}}{2^{2m}} \quad (19)$$

at the tiny cost of one bit of the additional mark-bit, which results in embedding rate decreased by $\frac{l}{l+1}$ in terms of percentage and extra modification quantity increased by 0.5. So, the embedding efficiency of the presented method here is EE, as formula (20) shown.

$$\mathrm{EE} = \frac{l}{\frac{l}{2} - \frac{C_{2m}^{m}}{2^{2m}} \times \frac{l}{2} + \frac{1}{2}} = \frac{2}{1 - \frac{C_{2m}^{m}}{2^{2m}} + \frac{1}{l}} \quad (20)$$

5 Experimental results and discussion

Simulation environment is Intel Core i5, 2.67 GHz CPU, 4 GB memory, and the compiler environment is Microsoft Visual Studio 2008, VC++6.0, in operating system Windows 7. Secret information is a binary bit stream generated by a pseudorandom generator.

The experimental images are downloaded from BOSSBase which are natural images of 5000 different content, hue, and texture complexity taken from different digital cameras. They are converted into a gray scale image of 512×512 in size.

Table 3 Performance comparison of HPSB and Mao's [11] algorithms

Embedding algorithm	Embedding rate	Embedding time (s/10^4 bit)	Embedding efficiency
Proposed algorithm	0.89	0.0023	2.35
Mao's [11] algorithm	0.75	0.0985	2.29

Table 1 lists the performance of the proposed steganography algorithm under the different block length l in the embedding process. The embedding rate and the distortion rate increase as l goes up as shown in Fig. 9a, b. Therefore, the improvement of distortion rate is at the expense of reducing embedding rate, and l can be used as a tradeoff control between embedding rate and distortion rate.

Compared with the LSB-based embedding algorithm, the signal-to-noise ratio is improved significantly. But with the change of block length l, the peak signal-to-noise ratio (PSNR) remains almost unchanged. Matrix embedding algorithms in [9, 26] improve embedding efficiency at considerable expense of reducing embedding rate. By contrast, the proposed HPSB algorithm well balances embedding rate and embedding efficiency. From experimental results, it is noted that with l increasing the embedding rate is improved while embedding efficiency remains 2.3 above as shown in Fig. 9c.

Under different block length of l, Table 2 compares the theoretical value of modification gain rate and embedding efficiency with the actual value. The actual value in the table is the average of the experimental data of the 5000 test images. From Table 2, it is an astonishing coincidence that the experimental data of modification gain rate and embedding efficiency are in line with the theoretical value.

It is revealed that in the course of embedding secret information, the change of the least significant bit is a discrete stationary stochastic process. Meanwhile, it is found that

Table 4 Performance comparison of HPSB and HPS

Image ID	PSNR		Histogram distortion		Relative entropy	
	HPS	HPSB	HPS	HPSB	HPS	HPSB
1	51.14	52.54	1172	282	0.000589	0.000093
2	51.14	52.53	786	360	0.000117	0.000080
3	51.14	52.53	618	374	0.000126	0.000059
4	51.14	52.53	952	384	0.000197	0.000097
5	51.15	52.52	602	320	0.000154	0.000055
6	51.16	52.54	752	388	0.000057	0.000029

this is an interesting case of the law of large numbers, i.e., Bernoulli theorem, in digital image processing.

All tests are carried out on approximately 5000 images with a fixed size of 512×512 pixels. The two average values of modification gain rate and embedding efficiency remain almost stable.

In addition, as formula (19) shows, the expression is some complicated and the computational complexity of MGR increases rapidly as m goes up, where the meaning of m is as same as formula (16). To be convenient for analysis, formula (19) is simplified to $\frac{1}{\sqrt{\pi m}}$ using the Stirling formula (17). From this simplified expression, it can be clearly noticed that the MGR slowly decreases with the increase of m as shown in Fig. 10. Under the experimental circumstance of this paper, that l is 8 can well balance embedding rate and histogram distortion.

The fast algorithm of matrix embedding proposed by Mao Qian in Ref. [11] is also based on minimizing the modification of pixel value to achieve higher embedding efficiency. In terms of embedding rate, embedding time, and embedding efficiency, the performance of HPSB algorithm is compared with that of [11] side by side in Table 3. In the comparison, the block length of the HPSB algorithm is 8. From the data in the table, it can be concluded that the embedding rate and embedding efficiency of HPSB steganography algorithm are higher than those of [11], other than the embedding time which is much less than that of [11]. In addition, the algorithm in [11] does not implement histogram preserving. Therefore, compared with the algorithm in [11], the proposed algorithm has the extra advantage of histogram preservation. As mentioned in Section 2.3, since algorithm in [28] is of high computational complexity, the corresponding comparison is omitted.

The data in Table 4 are the comparison between the HPS algorithm [33] and the proposed HPSB algorithm, in which the block length of the HPSB steganography algorithm is 8. The images used for comparison are shown in Fig. 11.

Relative entropy (RE) is defined as follows:

$$D(P_c \mid |P_s) = \sum\nolimits_{q \in Q} P_c(q) \log_2 \frac{P_c(q)}{P_s(q)} \tag{21}$$

where P_c and P_s are probability distribution of cover data and stego data respectively, and Q is the set of all possible values of cover data and stego data. For gray scale image, the formula can be described as:

$$D(P_c \mid |P_s) = \sum\nolimits_{n=0}^{255} P_c[n] \log \frac{P_c[n]}{P_s[n]} \tag{22}$$

Histogram distortion (HD) is defined as follows:

$$D_h = \sum_{n=0}^{255} \mid h_c[n] - h_s[n] \mid \tag{23}$$

where $h_c[n]$ and $h_s[n]$ respectively represent the number of cover pixels and stego pixels whose value is n.

The results in Table 4 demonstrate that the PSNR, RE, and HD of the HPSB steganography algorithm are superior to those of the HPS algorithm, but the HPSB

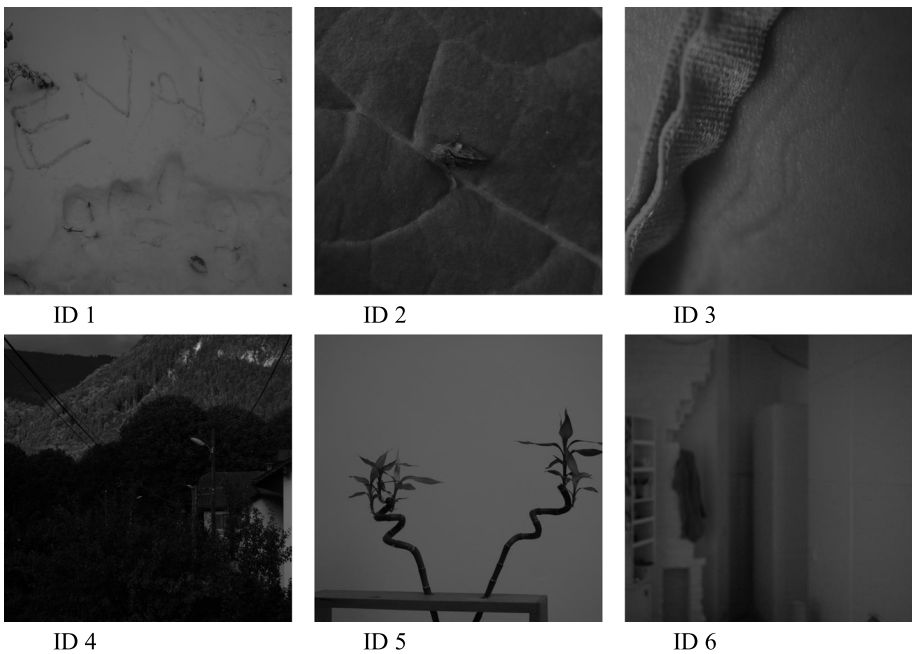

Fig. 11 The images in Table 4

steganography algorithm achieves this performance superiority at the cost of discounting the embedding capacity.

6 Conclusions

In this paper, the mathematical derivations of the HPSB algorithm are presented and the formulas exploited are verified by experimental data. Compared with other histogram feature-preserving steganography algorithms, the HPSB algorithm achieves higher peak signal-to-noise ratio, lower relative entropy, and lower histogram distortion, at the cost of insignificant reduction in embedding capacity. Compared with the given existing matrix embedding algorithms, with the same embedding capacity, HPSB algorithm attains higher embedding efficiency. Especially, when embedding capacity increases up to the upper limit, the embedding efficiency remains stable and the computational complexity stays low in HPSB. These advantages make HPSB steganography algorithm easier to use when applied on the devices with limited resources, e.g., mobile terminals. The further research is set to explore the adjustability of the embedding efficiency of HPSB algorithm. In addition, applying the proposed algorithm to video or audio domain is also worth investigating in the future.

Abbreviations

DR: Distortion rate; EE: Embedding efficiency; ER: Embedding rate; HD: Histogram distortion; HPS: Histogram-preserving steganography; HPSB: Histogram-preserving steganography based on block; LSB: Least significant bit; MGR: Modification gain rate; PCM: Parity check matrix; PSNR: Peak signal-to-noise ratio; RE: Relative entropy

Acknowledgements

This paper is sponsored by K.C.Wong Magna Fund in Ningbo University. Furthermore, the authors thank the editor and anonymous reviews for their helpful comments and valuable suggestions.

Funding

This work is partly supported by the National Natural Science Foundation of China (Grant No. U1736215, 61672302, 61300055), Zhejiang Natural Science Foundation (Grant No. LZ15F020002, LY17F020010), Ningbo Natural Science Foundation (Grant No.2015A610140), Ningbo University Fund (Grant No.XKXL1524, No. xkx11409, No.MNATKL2012002).

Availability of data and materials

Please contact author for data requests.

Authors' contributions

All authors took part in the work described in this paper. The team conducted literature reading and discussion together. The author Cheng Jie designed the proposed algorithm and the author Yang Rener made the theoretical derivation of mathematics. The author Chen Zhenzuo collected the test images from BOSSbase and preprocessed them, then Cheng Jie and Chen Zhenzuo together programed to implement and verify the proposed algorithm. The author Cheng Jie wrote the first version of this paper, and then Cheng Jie and Yang Rener repeatedly revised the manuscript. To accomplish the final manuscript submitted, all authors participated into discussion. All authors read and approved the final manuscript.

Authors' information

Cheng Jie was born on Jan 15, 1975, in Ningbo, Zhejiang. Current position: Lecturer in Ningbo University. University studies: Master degree in Computer Science and Technology from Zhejiang University in 2003. Scientific interest: information hiding, digital signal processing and machine learning. Chen Zhenzuo was born on Nov 10, 1992, in Ningbo, Zhejiang. University studies: Master's degree in Ningbo University, Bachelor's degree in China Pharmaceutical University. Scientific interest: information hiding, image processing, data analysis and deep learning. Yang Rener was born on Apr 2, 1968, in Ningbo, Zhejiang. Current position and grades: vice-professor in Ningbo University, PhD degree in Information and Communication Systems. Scientific interest: information hiding, image processing and image coding.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Received: 19 April 2018 Accepted: 10 July 2018
Published online: 22 August 2018

References

1. W Mazurczyk, K Szczypiorski, Trends in Steganography, Communications of the ACM, March 2014, Vol. 57 No. 3, Pages 86–95
2. J Fridrich, P Lisonek, Grid coloring in steganography. IEEE Trans. Inform. Theory **53**(4), 1547–1549 (2007)
3. HJ Kim, C Kim, Y Choi, S Wang, X Zhang, Improved modification direction methods. Computer & Math Application **60**(2), 319–325 (2010)
4. CN Yang, G-C Ye, C Kim, Data hiding in halftone images by XOR block-wise operation with difference minimization. KSII Trans Internet Info System **5**(2), 457–476 (2011)
5. Crandall R. Some Notes on Steganography, Posted on Steganography Mailing List, 1998.
6. J Fridrich, D Soukal, Matrix embedding for large payloads. IEEE Trans. Inf. Forensics Secur. **1**(3), 390–395 (2006)
7. A Westfeld, F5: A steganographic algorithm. Proceedings of the 4th international workshop information hiding. Lect. Notes Comput. Sci **2137**(1), 289–302 (2001)
8. W Luo, F Huang, J Huang, Edge adaptive image steganography based on LSB matching revisited. IEEE Trans. Inf. Forensics Secur. **5**(2), 201–214 (2010)
9. J Fridrich, M Goljan, P Lisonek, et al., Writing on wet paper. IEEE Trans. Inf. Forensics Secur. **53**(10), 3923–3935 (2005)
10. S Hetzl, P Mutzel, A graph–theoretic approach to steganography. Communications and Multimedia Security. CMS 2005. Lecture Notes in Computer Science, vol 3677, pp. 119-128. Springer, Berlin, Heidelberg
11. Q Mao, A fast algorithm for matrix embedding steganography. Digital Signal Process **25**, 248–254 (2014)
12. C Cachin, *An Information-Theoretic Model Steganography, IH98, LNCS 1525* (Springer Verlag, Heidelberg, 1998), pp. 306–318
13. T Yang, H Chen, Matrix embedding in steganography with binary reed-Muller codes. IET Image Process **11**(7), 522–529 (2017)
14. ZZ Gao, DW Wei, GM Tang, et al., Fast matrix embedding based on random linear code. ACTA Electronic SINICA **45**(5), 1139–1149 (2017)
15. C Kim, C-N Yang, Data hiding based on overlapped pixels using hamming code. Multimedia Tools Application **75**, 15651–15663 (2016)
16. A Sarkar, U Madhow, BS Manjunath, Matrix embedding with pseudo random coefficient selection and error correction for robust and secure steganography. IEEE Trans. Inf. Forensics Secur. **5**(2), 225–239 (2010)
17. CC Chang, TD Kieu, YC Chou, A high payload steganographic scheme based on (7,4) hamming code for digital images, proc. Int Symp Elec Comm and Sec **2008**, 16–21 (2008)
18. YK Gao, XL Li, TY Zeng, B Yang, *Improving Embedding Efficiency Via Matrix Embedding: A Case Study, Proc.16th IEEE International Conference on Image Processing* (ICIP, 2009), pp. 109–112
19. JY Chen, YF Zhu, Y Shen, WM Zhang, Efficient matrix embedding based on random linear codes. 2010 International Conference on Multimedia Information Networking and Security, Proc. MINES 2010, pp. 879–883, Dec. 2010

20. Y Gao, X Li, B Yang, Constructing specific matrix for efficient matrix embedding. 2009 IEEE International Conference on Multimedia and Expo, pp. 1006–1009, Jun. 2009
21. T Filler, J Judas, J Fridrich, Minimizing additive distortion in steganography using syndrome-trellis codes. IEEE Trans. Inf. Forensics Secur. **6**(3), 920–935 (2011)
22. C Kim, C-N Yang, Data hiding based on overlapped pixels using hamming code. Multimed Tools Appl **75**, 15651–15663 (2016)
23. T Hui, Q Jie, H Yong-feng, et al., Optimal matrix embedding for voice-over-IP steganography. Signal Process. **117**, 33–43 (2015)
24. W Chao, Z Wei-ming, L Jiu-fen, et al., Fast matrix embedding by matrix extending. IEEE Trans Inf Forens Secur **7**(1), 346–350 (2012)
25. G Liu, W Liu, Y Dai, et al., Adaptive steganography based on block complexity and matrix embedding. Multimedia Systems **20**, 227–238 (2014)
26. J Fridrich, M Goljan, R Du, Detecting LSB steganography in color, and gray-scale images. Multimedia IEEE **8**(4), 22–28 (2001)
27. J Bai, CC Chang, A high payload steganographic scheme for compressed images with hamming code. Int J Netw Sec **18**(6), 1122–1129 (2016)
28. X Zhang, S Wang, Efficient data hiding with histogram preserving property. Telecommunication System **49**, 179–185 (2012)
29. W Zhang, X Zhang, S Wang, in *Lecture Notes in Computer Science: Vol. 5284. Proceedings of the 10th Information Hiding Workshop*. Maximizing steganographic embedding efficiency by combining hamming codes and wet paper codes (Springer, Berlin, 2008), pp. 60–71
30. P Sallee, in *Lecture Notes in Computer Science: Vol. 2939. Proceedings of the 6th Information Hiding Workshop*. Model-based steganography (Springer, Berlin, 2004), pp. 154–167
31. X Zhang, S Wang, K Zhang, in *Lecture Notes in Computer Science: Vol. 2776. Computer Network Security*. Steganography with least histogram abnormality (Springer, Berlin, 2003), pp. 395–406
32. H Wu, J Dugelay, in *Lecture Notes in Computer Science: Vol. 5284. Proceedings of the 10th Information Hiding Workshop*. Cheung Y. A data mapping method for steganography and its application to images (Springer, Berlin, 2008), pp. 236–250
33. Yang R.E., Tao S., Zheng Z.W., et al., A digital image steganography method, Patent No.: ZL 2014 1 0277228.8 CN 104050624A.