Telecommun Syst (2012) 49:179–185
DOI 10.1007/s11235-010-9364-5

Efficient data hiding with histogram-preserving property

Xinpeng Zhang · Shuozhong Wang

Published online: 18 June 2010

Abstract For improving the steganographic security, the data-hiders always hope to lower the distortion level and to preserve the original data distribution. A novel efficient data hiding scheme with a histogram-preserving property is proposed in this work. After decomposing cover samples into a series of binary sequences, a number of candidate vectors representing each secret bit-block are produced, and a criterion for lowering steganographic distortion and keeping flip balance is set up for optimal selection from the candidates. A combination of flipping rates for different binary sequences to maximize the secret capacity is also studied. This way, the original data distribution in cover sequence is preserved, and a payload-distortion performance approaching the theoretical upper bound can be achieved.

Keywords Steganography · Histogram · Distortion · Embedding rate

1 Introduction

While digital steganography aims to embed secret messages into a carrier signal by altering the most insignificant components for covert communication, the purpose steganalysis is to detect the presence of hidden data by exploiting statistical abnormality caused by data embedding [1]. Generally speaking, the more the modification introduced into cover content, the more the hidden message is detectable to steganalytic attempts. For security enhancement, a data-hider always hopes to lower the level of distortion caused by data embedding and to preserve the statistical properties of the cover.

In order to reduce the data-embedding distortion, a series of steganographic coding methods based on binary system have been developed [2–4]. In these methods, a number of different patterns of cover data are used to represent an identical type of secret data, and the data-hider modifies the original cover data to the nearest pattern mapping the secret data to be hidden. When these methods is used in the least significant bit (LSB) plane of cover media, by flipping a small part of LSB from 0/1 to 1/0, a fairly large amount of secret data can be embedded. That means a high embedding efficiency. A great deal of efforts has been made to make the payload-distortion performance approach the theoretical limit [4]. However, the flipping probability of an LSB is independent of its original value. That implies if the LSBs having an original value 0/1 are more than those having an original value 1/0, the number of LSB flipped from 0/1 to 1/0 would be larger than that of LSB flipped from 1/0 to 0/1. As a result, the distribution of LSB will be changed, leading to vulnerability to steganalysis.

On the other hand, some steganographic methods capable of preserving original data distribution have been also presented. For example, Model-Based (MB) steganography [5] can ensure that the stego-data histogram is consistent with a model derived from the original histogram so as to avoid distribution abnormality. In this method, new data that represent the secret information and obey the distribution model are used to replace the original cover data. Using the techniques described in [6] and [7], the data-hider can embed the secret message into LSB of cover image when keeping the original histogram. However, the maximization of payload

X. Zhang (✉) · S. Wang
School of Communication and Information Engineering,
Shanghai University, Shanghai 200072, P.R. China
e-mail: xzhang@shu.edu.cn

S. Wang
e-mail: shuowang@shu.edu.cn

with a low distortion level is not concerned in these techniques. In other words, an efficient data-hiding method preserving the original histogram and keeping a low distortion level should be studied.

In this paper, we propose a novel steganographic scheme to maximize the payload when preserving a histogram of original cover data and constraining the steganographic distortion at a desired level. With this scheme, the cover samples are decomposed into a series of binary sequences, and an optimal one among a number of candidate stego-data for representing the secret information is chosen to control the distortion level while preserve the original distribution of each cover sequence. As a result, the amount of embedded data can approach the theoretical upper bound of steganographic capacity.

2 Steganographic framework

Using the following scheme, a data-hider can embed secret information into a cover signal while keep the original histogram unchanged and achieve a low distortion level. The framework of embedding/extracting process is described as follows.

(a) Before data embedding, the possible values of cover samples are divided into a series of bins, each of which contains two adjacent values. For instance, if pixels of an uncompressed gray image are used to carry secret information, there are 128 bins: $[\{0\ 1\}, \{2\ 3\}, \ldots, \{254\ 255\}]$. Alternatively, quantized DCT coefficients of a JPEG image are used, the bins are $[\ldots, \{-4\ -3\}, \{-2\ -1\}, \{0\ 1\}, \{2\ 3\}, \ldots]$ or $[\ldots, \{-3\ -2\}, \{-1\ 0\}, \{1\ 2\}, \{3\ 4\}, \ldots]$.

(b) Collect all the cover samples falling in each bin, and converts each sample into 0 or 1 according to the two different values in the bin, thus forming a binary sequence. The number of sequences is equal to that the number of bins. As an example, suppose the range of sample values is $[0, 3]$ that can be divided into two bins $\{0\ 1\}$ and $\{2\ 3\}$. Considering the cover $\{12023210221103202213\}$, one can obtain two subvectors $\{101011001\}$ and $\{22322232223\}$, which can be converted into binary sequences $\{101011001\}$ and $\{00100010001\}$.

(c) Modify the binary sequences to carry secret information. In the above example, assume that the two sequences are modified into $\{100011101\}$ and $\{01000011001\}$ after data-embedding. The data-hider may inversely convert them into two sequences $\{100011101\}$ and $\{23222233223\}$, and re-organize them as a stego-cover $\{12032200221113302213\}$ according to the original order. Here, the value of a cover sample is either kept unchanged or changed to another value belonging to a same bin.

(d) On the receiver side, after obtaining the stego-binary-sequences in a similar manner, the embedded secret information is extracted. The method for embedding secret data into a binary cover sequence and retrieving the embedded data from a stego-binary-sequence will be described in the next section.

2.1 Upper bound of embedding rate in a binary cover sequence

Considering a certain bin, let us denote the length of the corresponding binary cover sequence as N, and the rates of 0s and 1s as α and $(1-\alpha)$ respectively. Without loss of generalization, we assume $0 \le \alpha \le 0.5$. To keep the original data distribution, the number of bits changed from 0 to 1 in data-embedding should be equal to the number of bits changed from 1 to 0. Denote the number of bits to be changed as $2 \cdot \beta \cdot N$ ($\beta \le \alpha$), with β indicating the distortion level due to data embedding. Since the number of host bits with original value 0 is $\alpha \cdot N$, among which $\beta \cdot N$ bits are flipped, as explained in [3], the upper bound of the number of secret bits to be embedded into them is

$$B_+ = \alpha \cdot N \cdot H(\beta/\alpha) \tag{1}$$

where $H(\cdot)$ is the binary-entropy function

$$H(p) = -p \cdot \log_2 p - (1-p) \cdot \log_2 (1-p) \tag{2}$$

Similarly, the upper bound of the number of secret bits to be embedded in the host bits with original value 1 is

$$B_- = (1-\alpha) \cdot N \cdot H[\beta/(1-\alpha)] \tag{3}$$

Then, the upper bound of secret payload in a host bit-sequence is

$$B = B_+ + B_- \tag{4}$$

Consider the ratio between the number of embedded bits and the length of cover sequence, and call it the embedding rate. The upper bound of embedding rate is

$$B_R = B/N = \alpha \cdot H(\beta/\alpha) + (1-\alpha) \cdot H[\beta/(1-\alpha)] \tag{5}$$

For an extreme scenario, the data-hider replaces the original bit-sequence with a secret data sequence with a same 0–1 distribution. In this case, $\beta = \alpha \cdot (1-\alpha)$ and the secret payload is $N \cdot H(\alpha)$. That means the embedding rate is $H(\alpha)$, and the upper bound in (5) is achieved. Actually, $H(\alpha)$ is also the upper bound of embedding rate with β being larger than $\alpha \cdot (1-\alpha)$, since it is the entropy when the original binary distribution is kept unchanged. In the following, we only discuss the case of $\beta < \alpha \cdot (1-\alpha)$, which implies a lower distortion level.

2.2 Upper bound of embedding rate in binary cover-sequences

Denote the total number of bins as M, the lengths of the corresponding binary cover-sequences as $N_1, N_2, \ldots, N_M$, the rates of 0s in the cover-sequences as $\alpha_1, \alpha_2, \ldots, \alpha_M$, and the rates of 1s in the cover-sequences as $(1-\alpha_1)$, $(1-\alpha_2), \ldots, (1-\alpha_M)$, respectively. To keep the original data distribution, for each binary cover-sequence, the number of bits changed from 0 to 1 in data-embedding should be equal to the number of bits changed from 1 to 0. Denoting the number of bits to be changed in the m-th cover-sequence as $2 \cdot \beta_m \cdot N_m$ ($\beta_m \le \alpha_m, m = 1, 2, \ldots, M$), the energy of distortion caused by data embedding is

$$E = \sum_{m=1}^{M} 2 \cdot \beta_m \cdot N_m \tag{6}$$

With a given distortion level E, a data hider always hopes to maximize the upper bound of secret payload

$$B_T = \sum_{m=1}^{M} B_m \tag{7}$$

where

$$B_m = \alpha_m \cdot N_m \cdot H(\beta_m/\alpha_m) + (1-\alpha_m) \cdot N_m \cdot H[\beta_m/(1-\alpha_m)] \tag{8}$$

By applying Lagrange Multiplier method, for an extremum of B_T to exist, there must have

$$\frac{\partial B_T}{\partial \beta_1} \Big/ \frac{\partial E}{\partial \beta_1} = \frac{\partial B_T}{\partial \beta_2} \Big/ \frac{\partial E}{\partial \beta_2} = \cdots = \frac{\partial B_T}{\partial \beta_M} \Big/ \frac{\partial E}{\partial \beta_M} \tag{9}$$

That means

$$\frac{\beta_1^2}{(\alpha_1-\beta_1)\cdot(1-\alpha_1-\beta_1)} = \frac{\beta_2^2}{(\alpha_2-\beta_2)\cdot(1-\alpha_2-\beta_2)} = \cdots = \frac{\beta_M^2}{(\alpha_M-\beta_M)\cdot(1-\alpha_M-\beta_M)} \tag{10}$$

In other words, (10) is a necessary condition of maximizing the upper bound of secret payload. Defining a function

$$f(\alpha,\beta) = \frac{\beta^2}{(\alpha-\beta)\cdot(1-\alpha-\beta)} \tag{11}$$

we call a curve in the α–β plane as an α–β curve if the values of $f(\alpha,\beta)$ at the curve are same. Figure 1 shows three examples of $\alpha-\beta$ curves. So, after converting the cover samples into M binary sequences with various rates of 0s $\alpha_1, \alpha_2, \ldots,$ and α_M, the data hider should select a optimal combination of $\beta_1, \beta_2, \ldots,$ and β_M, which ensure that all the $(\alpha_1,\beta_1), (\alpha_2,\beta_2), \ldots,$ and (α_M,β_M) are within a same α–β curve, for different binary sequences to maximize the secret capacity with a given distortion level. Actually, the various α–β curves correspond to the different levels of distortion caused by data embedding.

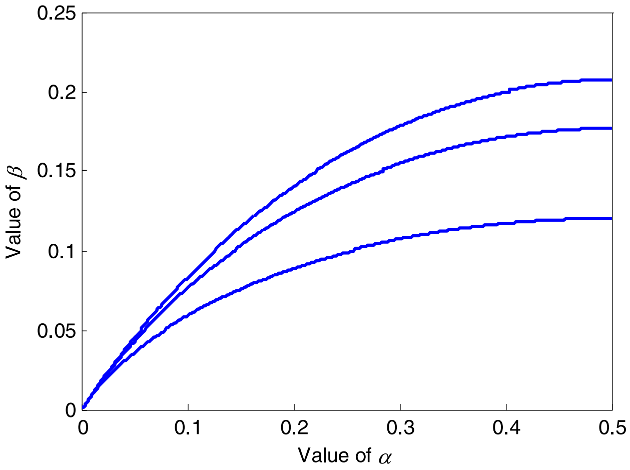

Fig. 1 Three examples of α–β curves, each of which having a same value of $f(\alpha,\beta)$

3 Efficient data-embedding in binary sequence

This section presents a data-embedding and data-extraction method in a binary cover sequence. In this method, a number of candidate stego-vectors representing a block of secret bits are produced, and a criterion for lowering steganographic distortion and keeping original distribution is used to select the optimal candidate. This way, the embedding rates can approach the upper bounds of (5).

3.1 Data embedding procedure

Assuming the secret message is a bit stream with uniform distribution, we divide the secret sequence into a series of bit blocks, each of which contains K bits. Generate a matrix $\mathbf{G}$ sized $16 \times (K+16)$, which is composed of two parts:

$$\mathbf{G} = [\mathbf{Q} \quad \mathbf{I}_{16}] \tag{12}$$

The left part $\mathbf{Q}$ sized $16 \times K$ is a pseudo-random binary matrix derived from a secret key, while the right part is a 16×16 identity matrix. For each secret bit-block $\mathbf{s} = [s_1, s_2, \ldots, s_K]$, we produce 2^{16} different types of binary vectors with a length $(K+16)$ using the following modulo-2 calculation,

$$\mathbf{v}_t = [s_1 \quad s_2 \quad \cdots \quad s_K \quad 0 \quad 0 \quad \cdots \quad 0] + [b(t,1) \quad b(t,2) \quad \cdots \quad b(t,16)] \cdot \mathbf{G} \quad t = 0, 1, \ldots, 2^{16} - 1 \tag{13}$$

The first vector on the right side of (13) is made up of **s** and 16 zeros, and the vector $[b(t,1), b(t,2), \ldots, b(t,16)]$ is a binary version of an integer t within $[0\ 2^{16} - 1]$,

$$b(t,u) = \lfloor t/2^{u-1} \rfloor \bmod 2, \quad u = 1, 2, \ldots, 16 \tag{14}$$

Denote the combination of an identity matrix sized $K \times K$ and the transpose of **Q** as **H**,

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_K & \mathbf{Q}^T \end{bmatrix} \tag{15}$$

Obviously, the modulo-2 product of **H** and the transpose of **G** is a zero matrix sized $K \times 16$. So, there must be

$$\mathbf{s} = \mathbf{H} \cdot \mathbf{v}_t^T \tag{16}$$

For each $\mathbf{v}_t$, we produce a corresponding candidate vector in the following steps.

(a) Calculate decimal value of $\mathbf{v}_t$,

$$v_t = \sum_{k=1}^{K+16} \mathbf{v}_t(k) \cdot 2^{k-1} \tag{17}$$

where $\mathbf{v}_t(k)$ is the k-th bit in the vector $\mathbf{v}_t$.

(b) Assign $y = 2^{(K+16)}$ and $x = v_t$, and let $\mathbf{c}_t$ be a vector with initialized length 0.

(c) Calculate

$$T = \max[1, \mathrm{round}(y \cdot \alpha)] \tag{18}$$

where the operator round(·) returns the nearest integer and max(·) returns the maximum value.

(d) If $x \le T - 1$, append a ‘0’ to the end of $\mathbf{c}_t$ and update the value of y

$$y \leftarrow T \tag{19}$$

Otherwise, append a ‘1’ to the end of $\mathbf{c}_t$ and update the values of y and x

$$y \leftarrow y - T, \qquad x \leftarrow x - T \tag{20}$$

(e) If $y > 1$, go to Step (c); otherwise, terminate the process.

We call the 2^{16} produced $\mathbf{c}_t$ as candidate vectors and denote their length as l_t. Here, probabilities of appending 0 and 1 are approximately α and $(1-\alpha)$, respectively. So, the rates of 0s and 1s in the candidate vectors would approximately be α and $(1 - \alpha)$. In other words, the data distribution in the candidate vectors is similar to that in the original binary cover sequence.

We embed the secret message into a binary cover sequence in a block-by-block manner:

(a) Denote the number of cover bits that have been flipped from 0 to 1 as S_0, and the number of cover bits flipped from 1 to 0 as S_1. Their initial values are both 0.

(b) Get a bit-block from the secret stream, and produce 2^{16} candidate vectors using the previously described method.

(c) For each candidate vector $\mathbf{c}_t$, compare it and the l_t bits in the cover sequence in a bit-by-bit manner. Denote the number of positions where the cover bit is 0 and the bit in $\mathbf{c}_t$ is 1 as $n_{t,0}$, and the number of positions where the cover bit is 1 and the bit in $\mathbf{c}_t$ is 0 as $n_{t,1}$. Calculate

$$r_t = n_{t,0} + n_{t,1} + \lambda \cdot (S_0 - S_1) \cdot (n_{t,0} - n_{t,1}) \tag{21}$$

(d) Find the smallest one among all $2^{16} r_t$,

$$v = \arg\min_t r_t \tag{22}$$

Modify the l_v bits in the cover sequence to make them the same as the corresponding $\mathbf{c}_v$. Update

$$S_0 \leftarrow S_0 + n_{v,0}, \qquad S_1 \leftarrow S_1 + n_{v,1} \tag{23}$$

(e) Repeat Steps (b)–(d) to embed other secret blocks into the rest of the cover sequence.

In (21), $n_{t,0} + n_{t,1}$ is the number of bit-alterations when modifying the l_t cover bits into a candidate vector $\mathbf{c}_t$, and $\lambda \cdot (S_0 - S_1) \cdot (n_{t,0} - n_{t,1})$ is a term of penalty related to the deviation from a balance between S_0 and S_1. The purpose of minimizing r_t is to ensure a low data-hiding distortion while keeping the original distribution of the cover sequence. If λ is too small, it is difficult to keep the balance between S_0 and S_1. On the other hand, if it is too large, the distortion level would be high. Based on a large number of experiments, we recommend to let $\lambda = 0.02$ for a satisfactory distortion level and flipping balance.

The level of embedding-induced distortion is determined by α and K. Table 1 lists the values of β with different α and K. With a given binary cover sequence, α is fixed. Then, the data-hider may select a suitable K to produce a stego-sequence with a desired β. Note that values of α and K should also be sent to the receiver using a different steganographic technique such as a plain LSB method. Although this reduces the number of cover bits for carrying the secret information, the influence on the overall performance is negligible since the additional payload is very small.

3.2 Data extraction procedure

Data extraction is easier than embedding. After obtaining a stego-sequence, the receiver first extracts the values of α and K, and then extracts the embedded data in the following steps.

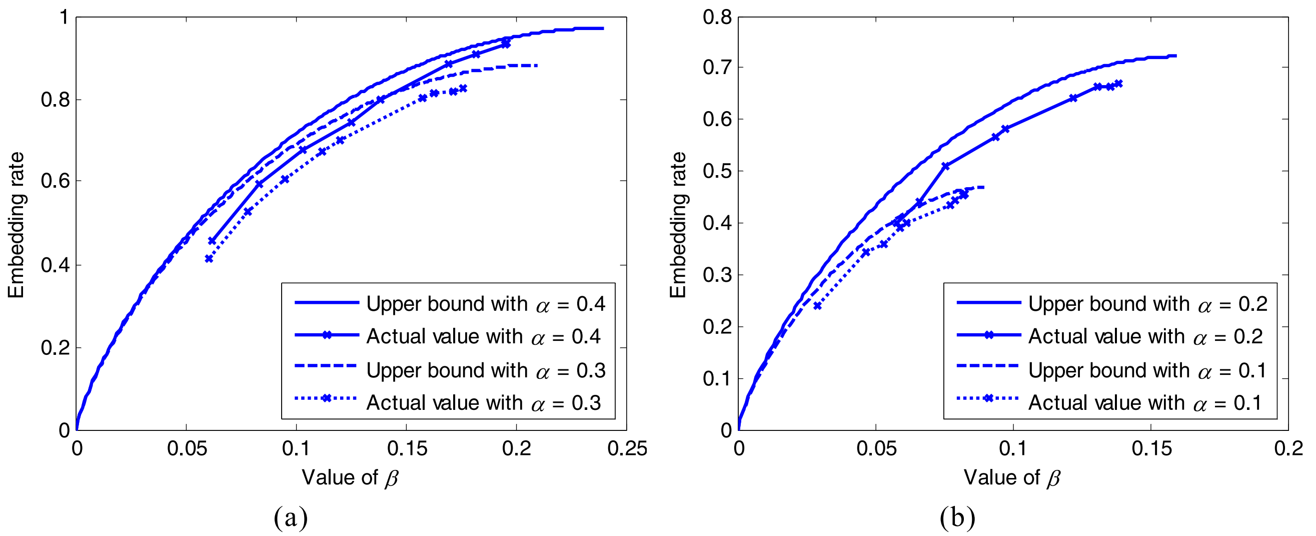

Fig. 2 Comparison between the actual embedding rates and the theoretical upper bounds: (**a**) $\alpha = 0.4$ or 0.3, and (**b**) $\alpha = 0.2$ or 0.1

Table 1 Values of β with different α and K

		K								
		15	25	35	60	100	200	300	400	500
α	0.1	0.029	0.047	0.053	0.059	0.062	0.077	0.079	0.082	0.083
	0.2	0.058	0.066	0.076	0.094	0.098	0.122	0.131	0.136	0.139
	0.3	0.061	0.078	0.095	0.112	0.120	0.158	0.163	0.172	0.176
	0.4	0.062	0.083	0.103	0.125	0.138	0.169	0.182	0.195	0.196

(a) Assign $y = 2^{(K+16)}$ and $x = 0$.

(b) Calculate

$$T = \max[1, \mathrm{round}(y \cdot \alpha)] \tag{24}$$

(c) Get a bit in the stego-sequence. If the encountered bit is 0, update the value of y

$$y \leftarrow T \tag{25}$$

Otherwise, update the values of y and x

$$y \leftarrow y - T, \qquad x \leftarrow x + T \tag{26}$$

(d) If $y > 1$, go to Step (b). Otherwise, convert x into $(K + 16)$ bits,

$$\mathbf{v}(k) = \lfloor x/2^{k-1} \rfloor \bmod 2, \quad k = 1, 2, \ldots, K + 16 \tag{27}$$

and calculate the secret bit-block

$$\mathbf{s} = \mathbf{H} \cdot \mathbf{v}^T \tag{28}$$

Then, go to Step (a) until all the bits in the stego-sequence are processed.

(e) Collect all the calculated secret bit-blocks and concatenate them to form the secret message.

3.3 Performance

In the experiments we embedded secret data with different values of α and K. When the number of flipped cover bits was more than 5000, difference between the numbers of cover bits flipped from 0 to 1 and flipped from 1 to 0 was always no more than 30. That means the data embedding method can sufficiently keep the original distribution of cover sequence. Figure 2 gives embedding rates and the theoretical upper bounds of (5). It can be seen that the embedding rates are very close to the corresponding bounds.

4 Experiment results

Using a 512×512 8-bit grayscale image "Lena" as the cover and converting all the pixels into 128 binary sequences according to the different bins $\{0\ 1\}, \{2\ 3\}, \ldots,$ and $\{254\ 255\}$, 99 binary sequences with lengths more than 100 were exploited to carry the secret data. Here, the values of β_m for the 99 different binary sequences satisfied (10), and the secret data were embedded by using the procedure described in Sect. 3.1. Figure 3 shows the original cover Lena and a stego-version with 2.5×10^5 secret bits embedded. In the

Fig. 3 (**a**) Original cover image Lena and (**b**) its stego-version with 2.5×10^5 secret bits embedded and PSNR 52.4 dB

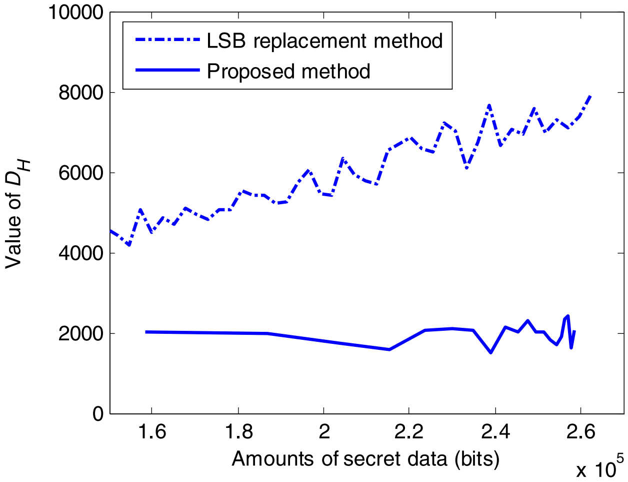

Fig. 4 Comparison of histogram changes between two embedding methods

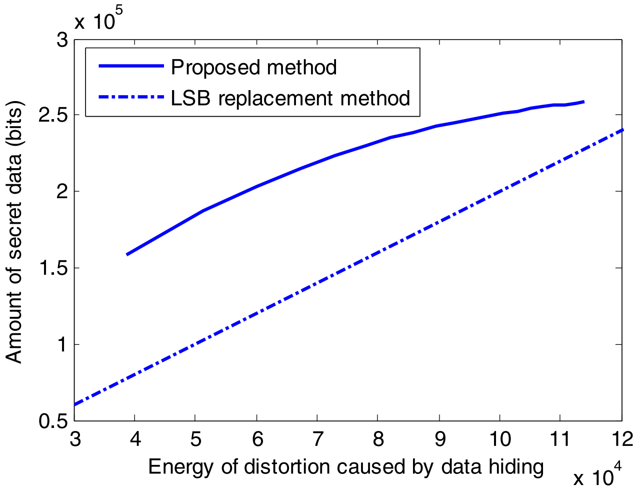

Fig. 5 Comparison of secret data amounts between two embedding with different distortion levels

stego-image, PSNR is 52.4 dB and the distortion is imperceptible.

Denoting the histograms of original image and stego-image as $[h_0, h_1, \ldots, h_{255}]$ and $[h'_0, h'_1, \ldots, h'_{255}]$, we can measure the histogram change caused by data hiding by using

$$D_H = \sum_{m=1}^{255} |h'_m - h_m| \tag{29}$$

Figure 4 compares the value of D_H of the traditional LSB replacement method and the proposed method when various amounts of secret data were embedded, while Fig. 5 compares the amount of embedded data of the two methods with different distortion levels. It can be seen that the proposed method has more embedding capacity and lower histogram distortion level.

5 Conclusion

This paper proposes a novel steganographic scheme with histogram-preserving and distortion-constraining properties, which can be employed for various cover samples such as pixels of uncompressed gray image and quantized DCT coefficients of JPEG image. After converting the cover samples into a series of binary sequences, a number of candidate stego-vectors representing a block of secret bits are produced. An exhaustive search is made to find the optimal candidate to replace the original cover bits. This way, the number of cover bits flipped from 0 to 1 approximately equals that flipped from 1 to 0, that is, the original distribution is kept, and the payload-distortion performance approaches the theoretical upper bound. In the future, we will try to further reduce computation complexity and increase the embedding rate.

Acknowledgement This work was supported by the Natural Science Foundation of China (60872116, 60832010, and 60773079), and the Key Scientific Research Project of Shanghai Education Committee (10ZZ59).

References

1. Wang, H., & Wang, S. (2004). Cyber warfare: steganography vs. steganalysis. *Communication of the ACM*, *47*(10), 76–82.
2. Zhang, X., & Wang, S. (2006). Dynamical running coding in digital steganography. *IEEE Signal Processing Letters*, *13*(3), 165–168.
3. Fridrich, J., & Soukal, D. (2006). Matrix embedding for large payloads. *IEEE Transformations on Information Forensics and Security*, *3*(1), 390–394.
4. Zhang, W., Zhang, X., & Wang, S. (2008). Maximizing steganographic embedding efficiency by combining hamming codes and wet paper codes. In *Lecture notes in computer science: Vol. 5284*. *Proceedings of the 10th information hiding workshop* (pp. 60–71). Berlin: Springer.
5. Sallee, P. (2004). Model-based steganography. In *Lecture notes in computer science: Vol. 2939*. *Proceedings of the 6th information hiding workshop* (pp. 154–167). Berlin: Springer.
6. Zhang, X., Wang, S., & Zhang, K. (2003). Steganography with least histogram abnormality. In *Lecture notes in computer science: Vol. 2776*. *Computer network security* (pp. 395–406). Berlin: Springer.
7. Wu, H., Dugelay, J., & Cheung, Y. (2008). A data mapping method for steganography and its application to images. In *Lecture notes in computer science: Vol. 5284*. *Proceedings of the 10th information hiding workshop* (pp. 236–250). Berlin: Springer.

Xinpeng Zhang received the B.S. degree in computational mathematics from Jilin University, China, in 1995, and the M.E. and Ph.D. degrees in communication and information system from Shanghai University, China, in 2001 and 2004, respectively. Since 2004, he has been with the faculty of the School of Communication and Information Engineering, Shanghai University, where he is currently a Professor. His research interests include information hiding, image processing and digital forensics.

Shuozhong Wang received B.S. degree in 1966 from Peking University, P.R. China, and Ph.D. degree in 1982 from University of Birmingham, England.

He was with Institute of Acoustics, Chinese Academy of Sciences, from January 1983 to October 1985 as research fellow, and joined Shanghai University of Technology in October 1985 as associate professor. He is now professor of School of Communication and Information Engineering, Shanghai University. He was associate scientist at Department of EECS, University of Michigan, USA, from March 1993 to August 1994, and a research fellow at Department of Information Systems, City University of Hong Kong, in 1998 and 2002.

His research interests include underwater acoustics, image processing, and multimedia security. He has published more than 150 papers in these areas. Many of his research projects are supported by the Natural Science Foundation of China.