# A Data Mapping Method for Steganography and Its Application to Images

Hao-tian Wu[1], Jean-Luc Dugelay[1], and Yiu-ming Cheung[2]

[1] Department of Multimedia Communication, Eurecom Institute,
Sophia Antipolis, France.
E-mail: {Haotian.Wu, Jean-Luc.Dugelay}@eurecom.fr
[2] Department of Computer Science, Hong Kong Baptist University,
Hong Kong SAR, China.
E-mail: ymc@comp.hkbu.edu.hk

**Abstract.** In this paper, a new steganographic method that preserves the first-order statistics of the cover is proposed. Suitable for the passive warden scenario, the proposed method is not robust to any change of the stego object. Besides the relative simplicity of both encoding and decoding, high and adjustable information hiding rate can be achieved with our method. In addition, the perceptual distortion caused by data embedding can be easily minimized, such as in the mean squared error criterion. When applied to digital images, the generic method becomes a sort of LSB hiding, namely the LSB$^+$ algorithm. To prevent the sample pair analysis attack, the LSB$^+$ algorithm is implemented on the selected subsets of pixels to preserve some important high-order statistics as well. The experimental results of the implementation are promising.

**Keywords:** Steganography, LSB$^+$ algorithm, bijective mapping, first-order statistics, sample pair analysis.

## 1 Introduction

The art of steganography, i.e. covert communication by hiding the presence of a message from a third party, has been studied in the community (e.g. [1]-[3]). Although the early steganographic methods can imperceptibly embed data into a cover object, the technique of steganalysis [4] has been developed to detect the hidden data from the statistical characteristics of the stego object. It has been shown by the detection-theoretic analysis (e.g. [5, 6]) that several data hiding methods are detectable. How to avoid being detected by the steganalysis technique is a central topic of the steganography research.

Since most of the steganalytic algorithms (e.g. [7]-[16]) exploit the statistics of the stego object for detection, quite a few steganographic algorithms (e.g. [17]-[23]) are designed to preserve the statistics of the cover object as much as possible. An early attempt is the F5 algorithm [17], in which some statistical characteristics in the histogram of DCT coefficients is preserved to prevent the

$\chi^2$ (chi-squared) attack [7]. In the detector designed by Fridrich et al. [8] to break the F5 steganography, the cover histogram is estimated from the suspected image for comparison. In Provos' Outguess [18], part of the JPEG coefficients are used to repair the histogram changed by data embedding. However, the changes at the JPEG block boundaries can be exploited because the embedding is performed in the block-wise transform domain [9]. A method attempting to preserve the histogram after LSB hiding is further presented by Franz [19], where a message that mimics the imbalance between the adjacent histogram bins is embedded in the pairs of values that are independent. Despite that a message with the unequal probabilities of 0 and 1 carries less information, the asymmetric embedding process determined by a co-occurrence matrix can be exploited for steganalytic attack, as shown in [10]. Similarly, Eggers et al. propose a histogram-preserving data-mapping (HPDM) method [20] by embedding a message with the same distribution as the cover object. Subsequently, the histograms of the cover object and the stego object can be matched so as to reduce the probability of being detected. However, it is shown by Tzschoppe et al. [21] that the HPDM can be detected by Lyu and Farid's steganalytic method [12] based on the high-order statistics. The reason given in [21] is that the higher frequency components have not been separately treated from the lower and direct current (DC) ones. In [22], a histogram restoration algorithm is proposed without embedding in the low-probability region. Within the embedding positions specified a secret key, a portion of eligible coefficients are used for embedding while the rest are used for compensation. In [23], the statistical restoration method is adopted to further preserve the second-order statistics of the cover image.

The model-based method [24] provides a new perspective for steganography by generating the stego object conforming to a given distribution model. For the lack of a perfect model, the steganographic algorithm using the Generalized Cauchy distribution [24] can be broken by only using the first-order statistics, i.e. the measures without considering the inter-dependencies between observations, such as mean and variance [25]. In this paper, a new steganographic method is proposed to preserve the first-order statistics inherently. By dividing the distribution range of the elements in a cover object into non-overlapped bins, two adjacent ones are utilized to form an individual embedding unit. Then the elements in the same embedding unit are bijectively mapped to each other for data embedding. Provided that the stego object is intact, the hidden message can be correctly extracted. Despite the relative simplicity of both encoding and decoding, high and adjustable information hiding rate can be achieved. Moreover, the distortion can be easily minimized in the minimum mean square error (MSE) criterion. When applied to digital images, the generic method becomes a sort of LSB hiding, namely the LSB$^+$ algorithm. To avoid being detected by the sample pair analysis (SPA) steganalysis [11], the LSB$^+$ algorithm is implemented on the subsets of pixels with the same neighbor values (up, down, left and right) to preserve some important high-order statistics as well.

The rest of this paper is organized as follows: In the next section, a novel data mapping method is presented for steganography. In Section 3, we apply it

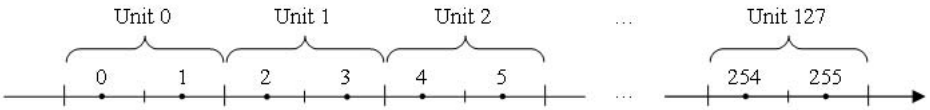



**Fig. 1.** Every two adjacent bins within the range from 0 to 255 are utilized to form an embedding unit for digital gray-scale images, respectively.

to digital images and further prevent the SPA attack by implementing it on the selected subsets of pixels. The performance of the new approach is evaluated in Section 4. Finally, a conclusion is drawn in Section 5.

## 2 A Data Mapping Method for Steganography

In this section, a novel LSB hiding algorithm named LSB$^+$ is firstly introduced, which preserves the image histogram. Then the generic data mapping method is further proposed, applicable to the cover object represented by integers or floating point numbers. We further analyze the bounds of information hiding rate and perceptual distortion with the proposed method.

### 2.1 LSB$^+$ Algorithm

In [3], Cachin proposes an information-theoretic model for steganography with the relative entropy, also called the Kullback-Leibler (K-L) divergence, between the distribution $P_C$ according to which the cover object is generated and the distribution $P_S$ corresponding to the stego object:

$$D(P_C||P_S) = \sum P_C \log \frac{P_C}{P_S}. \tag{1}$$

In general, $D(P_C||P_S)$ is nonnegative and equal to zero if and only if $P_C = P_S$. As for digital images, the high-order statistics can still be exploited for steganalysis after the cover histogram is preserved. Nevertheless, we regard it as a necessary condition for a secure image steganography. In the following, a novel LSB hiding algorithm named LSB$^+$ is developed to preserve the image histogram, as well as the other first-order statistics:

Given a gray-scale image, we can easily calculate its histogram by counting the pixels having the same value, i.e. the amount of pixels within the same bin. As shown in Fig. 1, every two adjacent bins within the range from 0 to 255 are utilized to form an embedding unit, respectively. We restrict the change of a pixel value within each unit so that only the least significant bit is changeable. For example, a pixel with the value of 4 can only be modified to 5 or remain the same because only the two pixel values 4 and 5 are contained in the same unit. Since the operations in one embedding unit are independent from those in the other units, we only discuss the operations in an arbitrary unit.

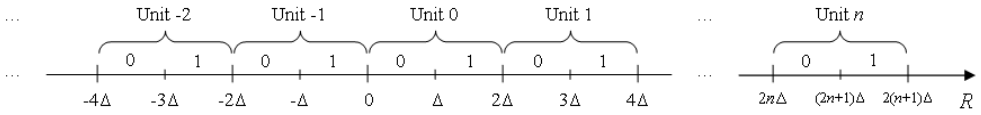



**Fig. 2.** Every two adjacent bins with the size $\Delta$ form an individual unit in the proposed data mapping method, respectively.

In the normal LSB hiding, a string of bit values are used to replace the original LSBs of pixel values. The histogram of the cover image is probably changed due to the randomness of the embedded data. Obviously, the histogram can be preserved if the amount of pixels within each bin is unchanged. In the LSB$^+$ algorithm, the bit values are also embedded by replacement but the replacement operations are performed conditionally. The key idea is that the number of the embedded 0s and 1s should not exceed the original ones in the LSBs, respectively. Suppose that there are $L$ and $M$ pixels originally in the left and right bins, the time of embedding 0 should be no more than $L$ and the time of embedding 1 should not exceed $M$. Once there are $L$ 0s (or $M$ 1s) having been embedded, all the unprocessed LSBs will be replaced with 1s (or 0s). In this way, the amounts of 0s and 1s in the LSBs are unchanged by data embedding. In the decoding process, the embedded bits are extracted one by one in the same order as in the embedding process. For each unit, the extraction process is finished once all the LSBs in either bin have been retrieved.

Since part of the LSBs are replaced to repair the cover histogram instead of embedding, the LSB$^+$ algorithm is a bit more complex than the normal LSB hiding. A portion of payload is also sacrificed to preserve the image histogram, as well as the other low-order statistics. In the following, a generic method that is applicable to any cover object represented by floating point numbers or integers will be further proposed.

### 2.2 The Generic Method

Suppose a cover object $\mathbf{C}$ consists of $N$ data elements, i.e. $\mathbf{C} = \{e_1, e_2, \cdots, e_N\}$, where $e_i$ is a data element with an index number $i \in \{1, 2, \cdots, N\}$. We use $\mathbf{R}$ to denote the distribution range of the data elements $\{e_1, e_2, \cdots, e_N\}$ and quantize $\mathbf{R}$ into the non-overlapping bins with the same size $\Delta$. For the sake of simplicity, we only discuss the one-dimensional case because multiple dimensions can be addressed one by one. As shown in Fig. 2, every two adjacent bins in the range of $\mathbf{R}$ form an individual unit, within which the bit values 0 and 1 are assigned to the left and right bins, respectively. If the value of a data element $e_i$ falls into the left bin, it represents a bit value of 0, or 1 if it is in the right bin. To embed a bit value of 0, the data element should be kept in the left bin if it was originally the case, or mapped to the left bin if it originally was in the right one. The process to embed a bit value of 1 is similar as long as we replace "left" by "right" and vice versa. The key idea of the proposed method is that the times

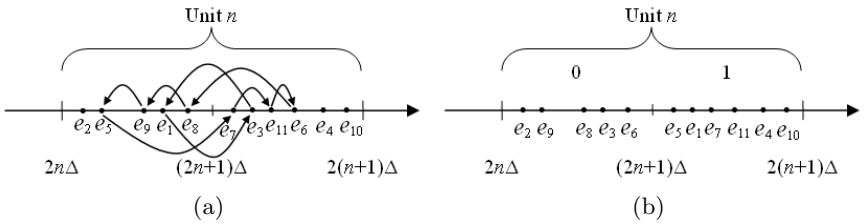



**Fig. 3.** The eleven data elements $\{e_1, e_2, \cdots, e_{11}\}$ in the embedding Unit $n$ are used to embed a string of bit values "10011010010". Only the first nine bit values "100110100" can be embedded until the time of embedding a bit value 0 has reached the amount of those elements originally in the left bin. Then the bijective mapping between the eleven elements are performed with the minimum mean square error (MSE).

of embedding 0 (1) should not exceed the amounts of elements originally in the left (right) bins, respectively. Therefore, we need to count the numbers of data elements in both bins before and during the embedding process. Once the time of embedding 0 (or 1) has reached the amount of elements originally in the left (or right) bin, no bit value can be further embedded to ensure that all elements in an embedding unit can be bijectively mapped to each other.

The detailed data mapping process can be illustrated in Fig. 3, where there are eleven data elements $\{e_1, e_2, \cdots, e_{11}\}$ with different values in the Unit $n$. To embed a string of bit values "10011010010", the data elements are processed in the order of their indices. Since $e_1$ is in the left bin, it corresponds to the bit value 0. Therefore, it should be mapped into the right bin to embed a bit value 1. As for $e_2$, it should remain in the left bin to embed a bit value 0. To embed the third bit value 0 in the string, $e_3$ needs to be mapped from the right to the left bin. The rest of the bit values are sequentially embedded until the ninth one, which leads $e_9$ to remain in the left bin. Since the number of the elements mapped to the left bin has reached 5, which is the amount of those originally in the left bin, no bit value can be embedded in the Unit $n$ any more due to the randomness of data to be embedded. Therefore, only the first nine bit values "100110100" can be embedded by mapping the data elements with the indices 2, 3, 6, 8, 9 into the left bin and the rest elements into the right bin. To minimize the error caused by data mapping in the mean square error (MSE) criterion, the elements mapped to the same bin should be ordered according to their original values. In the optimal scheme, $e_2, e_9, e_8, e_3, e_6$ are mapped to the data elements $e_2, e_5, e_9, e_1, e_8$ while the elements with the indices 5, 1, 7, 11, 4, 10 are mapped to those with the indices 7, 3, 11, 6, 4, 10. It should be noted that the data mapping process can be performed no matter whether several elements have the same values (e.g. the pixels having the same value in a gray-scale image). If all elements originally in a destination bin have the identical values, there is no need to order the ones mapped to that bin.

The data mapping between the data elements in an embedding unit heavily depends on the order they are processed. In Fig. 4, the same data elements as

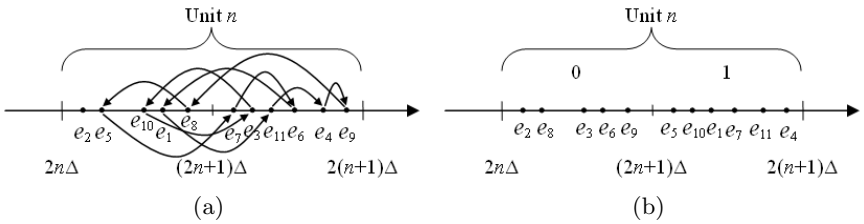



**Fig. 4.** The same data elements as shown in Fig. 3 are used to embed a string of bit values "100110100" except that the indices of the ninth and tenth elements are exchanged. As a result, the data mapping with the minimum MSE is greatly different from that in Fig. 3.

shown in Fig. 3 are used to embed the bit values "100110100" except that the indices of the ninth and tenth elements are exchanged. To embed the ninth bit value 0, the data element $e_9$ in Fig. 4 should be mapped from the right bin to the left one. In contrast, the data element $e_9$ in Fig. 3 remains in the left bin. To minimize the error in the MSE criterion, the data elements $e_2, e_8, e_3, e_6, e_9$ are mapped to the elements $e_2, e_5, e_{10}, e_1, e_8$, while the data elements with the indices 5, 10, 1, 7, 11, 4 are mapped to those with the indices 7, 3, 11, 6, 4, 9.

The decoding process is much simpler: Given that the order of data elements in the stego object is the same as that in the cover object, the bit values can be extracted from the positions of data elements (i.e. in the left or right bin) one by one. The extracted bit value will be 0 if a data element is located in the left bin, or 1 if it is in the right one. For each embedding unit, once all elements in either bin (left or right) have been used up for data extraction, the extraction process is finished. For example, the bit values that can be extracted from the Unit $n$ in Fig. 3 (b) and Fig. 4 (b) are not "10011010011", but "100110100". Since the embedding and extraction operations within each unit do not interfere with those performed in other units, the operations in every embedding unit can be carried out in parallel. So both of the encoding and decoding processes are performed in the order of all elements in a cover object. Furthermore, the order can be scrambled with a secret key shared by the sender and receiver.

### 2.3 Bounds of Hiding Rate and Perceptual Distortion

For each embedding unit, the amount of bit values that can be embedded depend on the amount of data elements in the two bins, respectively. Suppose there are $L$ and $M$ data elements in the two bins. Without loss of generality, we assume that $M$ is always no more than $L$. Then the minimum and maximum amount of bit values that can be embedded in that embedding unit are $M$ and $L + M - 1$. The upper bound of capacity is possible to be approached when $M$ is close to $L$ while the low bound is likely when $M$ is close to 0. In particular, the capacity will be zero when $M = 0$. If we take digital images for instance, the proposed method

tends to embed more when the histogram of the cover image changes slowly and the data hiding rate drops when the cover histogram fluctuates rapidly.

The data hiding rate is maximized by default because the embedding process will not stop until the bit values have been embedded to all elements in either bin (left or right). Alternatively, the hiding rate can be adjusted with a parameter $\theta \in (0, 1]$, i.e. once the time of embedding a bit value 0 (or 1) reaches a fraction (denoted by $\theta$) of the amount of elements originally in the left (or right) bin, the embedding process will be finished. Accordingly, the same policy is enforced in the extraction process. So the low and upper bounds of the data hiding capacity in the aforementioned embedding unit are $\lceil M\theta \rceil$ and $\lceil (L + M - 1)\theta \rceil$ bits, where $\lceil \cdot \rceil$ represents the ceil function. In this way, the data hiding rate can be adjusted with the parameter $\theta$, which should be shared by the sender and the receiver.

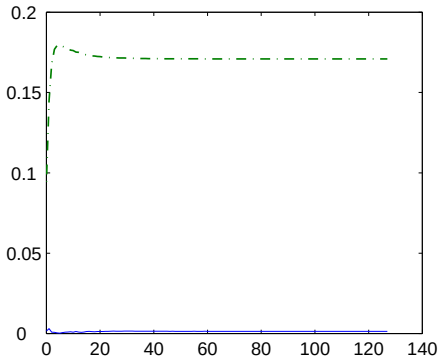
By performing the bijective mapping between the data elements within two adjacent bins, the perceptual distortion caused by data embedding is bounded. Given a bin size $\Delta$, the maximum change of a data element is always less than $2\Delta$. So the perceptual distortion of the stego object can be tuned by adjusting the bin size $\Delta$. The proposed method can be applied to the cover object represented by integers or floating point numbers. As for the floating point numbers, there is no need to deal with the truncation error as no new value is generated in the stego object. In this paper, we concentrate on image steganography by applying the LSB$^+$ algorithm, which is a specific case of the proposed method applied to images with the bin size set to 1.

## 3 Image Steganography with the LSB$^+$ Algorithm

Since there is only one pixel value in each bin as shown in Fig. 1, there is no need to order the pixels mapped to the same bin. We perform the LSB$^+$ algorithm on all pixels within a cover image in the raster order, i.e. by rows from top to bottom and within each row from left to right. By setting the parameter $\theta$ to 1, the stego image is generated with the hiding rate at 0.9688 bit/pixel and $PSNR = 51.14dB$, as shown in Fig. 5 (a). Since the LSB$^+$ algorithm preserves the histogram, the steganalytic algorithms based on histogram are no longer efficient. Furthermore, it is performed in the spatial domain without differentiating the pixels at the block boundaries and those within the blocks. So the steganalytic algorithms designed to detect the message in a specific transform domain (e.g. JPEG) or a block structure are incapable of detection. Nevertheless, readers may argue that the hidden message may be detected by the steganalytic algorithms using high-order statistics (e.g. [6], [11]-[16]). In the following, we further take the SPA attack in [11] for instance and explore the inter-dependencies between pixels to prevent it.

Dumitrescu et at. develop the technique of SPA in [11] to detect the random LSB hiding in digital images. The key assumption for the SPA steganalysis can be summarized as follow: For the sampled pairs of pixels whose values differ by an odd number, the chances that the greater pixel value is odd or even are equal. The closed multi-set $C_m$ under the LSB hiding is defined as the set of

(a) The stego image of "lena" with the hiding rate at 0.9688 bit/pixel and $PSNR = 51.14dB$.




(b) The relative errors calculated from the original and stego images of "lena" as shown by the solid and dash-dot curves, respectively.

**Fig. 5.** The relative error $\frac{||\bigcup_{m=0}^{j} Y_{2m+1}|-|\bigcup_{m=0}^{j} X_{2m+1}||}{|\bigcup_{m=0}^{j} Y_{2m+1}|+|\bigcup_{m=0}^{j} X_{2m+1}|}$ is greatly increased for $0 \leq j \leq 127$ after implementing the LSB$^+$ algorithm on all pixels in the cover image of "lena" with $\theta = 1$.

pixel pairs whose values differ by $m$ in all the bits except the least significant one (i.e. by right shifting one bit to get rid of the LSB). But its submultisets $D_{2m}$ (the set of pixel pairs whose value differ by $2m$), $X_{2m-1}$ (the set of pixel pairs whose values differ by $2m-1$ and the greater value is even), and $Y_{2m+1}$ (the set of pixel pairs whose values differ by $2m+1$ and the greater value is odd), are not close under the LSB hiding. As shown in Fig. 5 (b), the relative error $\frac{||\bigcup_{m=0}^{j} Y_{2m+1}|-|\bigcup_{m=0}^{j} X_{2m+1}||}{|\bigcup_{m=0}^{j} Y_{2m+1}|+|\bigcup_{m=0}^{j} X_{2m+1}|}$ is greatly increased after implementing the LSB$^+$ algorithm on all pixels in the cover image of "lena" with $\theta = 1$, where $|X_{2m-1}|$ and $|Y_{2m+1}|$ denote the amount of pixel pairs in $X_{2m-1}$ and $Y_{2m+1}$, respectively. The phenomenon is modeled by a finite-state machine in [11]. To further estimate the length of message embedded by the random LSB hiding, the fraction of the pixels modified in the embedding process is assumed to be equal to $\frac{p}{2}$ when the data hiding rate is $p$ bit/pixel. However, the same conclusion cannot be drawn from the LSB$^+$ algorithm because part of the pixel values are modified not for data embedding purpose but to preserve the histogram of the cover. So we directly use $\alpha$ to denote the fraction of the pixels modified in the embedding process. Then the fraction of the pixels that are unchanged is $1-\alpha$. For $m = 1, 2, \ldots, 127$, (2) and (3) in [11] become

$$|X_{2m-1}|(1-2\alpha)^2 = \alpha^2|C_m| - \alpha(|D'_{2m}| + 2|X'_{2m-1}|) + |X'_{2m-1}|, \qquad (2)$$

$$|Y_{2m+1}|(1-2\alpha)^2 = \alpha^2|C_m| - \alpha(|D'_{2m}| + 2|Y'_{2m+1}|) + |Y'_{2m+1}|, \qquad (3)$$

where $|C_m|$ denotes the amount of pixel pairs in $C_m$. $|X'_{2m-1}|$ and $|Y'_{2m+1}|$ denote the amount of pixel pairs whose values differ by $2m-1$ while the greater value is even and odd in the samples from the stego image, respectively. $|D'_{2m}|$ denotes the amount of pixel pairs whose values differ by $2m$ in the samples from the stego image. When $m = 0$, the (4) in [11] becomes

$$|Y_1|(1-2\alpha)^2 = 2\alpha^2|C_0| - 2\alpha(|D'_0| + |Y'_1|) + |Y'_1|. \tag{4}$$

In [11], $|X_{2m+1}|$ is assumed to be equal to $|Y_{2m+1}|$ for $m = 0, 1, \ldots, 127$. With this assumption, we can obtain the following quadratic equation to estimate the length of the hidden message

$$(|C_m|-|C_{m+1}|)\alpha^2-(|D'_{2m}|-|D'_{2m+2}|+2|Y'_{2m+1}|-2|X'_{2m+1}|)\alpha+|Y'_{2m+1}|-|X'_{2m+1}| = 0 \tag{5}$$

for $m \geq 1$ and for $m = 0$,

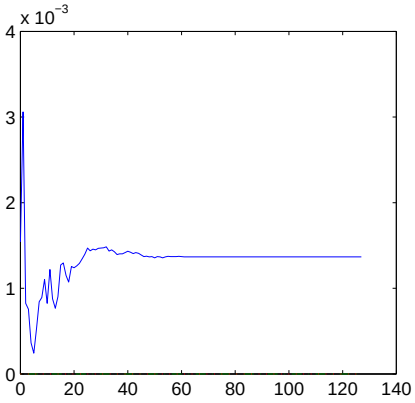
$$(2|C_0| - |C_1|)\alpha^2 - (2|D'_0| - |D'_2| + 2|Y'_1| - 2|X'_1|)\alpha + |Y'_1| - |X'_1| = 0. \tag{6}$$

It has been shown in [11] that the length of the hidden message is the smaller root of (5) provided that $|C_m| > |C_{m+1}|$ (or $2|C_0| > |C_1|$ for (6)). However, if $|X_{2m+1}| = |X'_{2m+1}|$ and $|Y_{2m+1}| = |Y'_{2m+1}|$, $|X'_{2m+1}| - |Y'_{2m+1}|$ in (5) is equal to 0 so that the estimated length will be zero. In the following, we will show how to prevent the SPA attack by implementing the LSB$^+$ algorithm on every special set of pixels.

For the better estimation, the SPA steganalysis is usually performed on the neighboring pixels to utilize the inter-dependencies between them. Since every sampled pixel pair are two neighboring pixels, we choose to implement the LSB$^+$ algorithm on the subset of pixels having the same neighbor values, i.e. the up, down, left, and right neighbor values (denoted by 4-N in short) of all pixels in the subset are the same. To generate a special subset, half of the pixel values, which are the neighbor values of the other pixels in a gray-scale image, are fixed during the embedding process. For each combination of the four neighbor values that appears, we count its occurrence in the first half pixel values and those pixels within them are grouped to a subset. Then the LSB$^+$ algorithm is implemented on every generated subset. By this means, the stego image of "lena" is generated with 5564 bits embedded and $PSNR = 65.17dB$, as shown in Fig. 6 (a). It should be noted that the subsets of pixels with the same neighbor values can exactly be generated from the stego image.

The effects of data embedding on the sampled pixel pairs are compensated by each other after implementing the LSB$^+$ algorithm on every subset of pixels having the same neighbor values. Consider a pixel $P_i$ whose LSB has been changed from 0 to 1, its value is changed from $2n$ to $2n+1$ with $n \in \{0, 1, \ldots, 127\}$. As the histogram is unchanged by the LSB$^+$ algorithm, there exists a corresponding pixel $P_j$ whose neighbor values are the same as $P_i$'s and its value has been changed from $2n+1$ to $2n$. Given a neighbor value $V_k$ of $P_i$ and $P_j$, it is unchanged during the embedding process. So the difference between the value of $P_i$ and $V_k$ will increase by 1 after the value of $P_i$ is changed from $2n$ to $2n+1$ if

(a) By applying the LSB$^+$ algorithm on every subset of pixels having the same four neighbor values (up, down, left, and right), the stego image is generated with 5564 bits embedded and $PSNR = 65.17dB$.




(b) Solid curve: Relative error of the cover image "lena", which is the same as the solid curve in Fig. 5 (b); Lines at the bottom: $|X'_{2m+1}| - |X_{2m+1}|$ and $|Y'_{2m+1}| - |Y_{2m+1}|$, which are zeros for $0 \leq m \leq 127$ so that the relative error of stego image is the same as the cover one.

**Fig. 6.** By implementing the LSB$^+$ algorithm in the 4-N way, the relative error $\frac{||\bigcup_{m=0}^{j} Y_{2m+1}| - |\bigcup_{m=0}^{j} X_{2m+1}||}{|\bigcup_{m=0}^{j} Y_{2m+1}| + |\bigcup_{m=0}^{j} X_{2m+1}|}$ of the cover image is unchanged.

$V_k \in [0, 2n]$, and the difference between the value of $P_j$ and $V_k$ will decrease by 1 after the value of $P_j$ is changed from $2n+1$ to $2n$. When $V_k \in [2n+1, 255]$, the difference between the value of $P_i$ and $V_k$ will decrease by 1 after the value of $P_i$ is changed from $2n$ to $2n+1$, and the difference between the value of $P_j$ and $V_k$ will increase by 1 after the value of $P_j$ is changed from $2n+1$ to $2n$. As a result, $|X_{2m+1}|$ and $|Y_{2m+1}|$ will be unchanged by the embedding process for $m = 0, 1, \ldots, 127$. As shown in Fig. 6 (b), the values of $|X'_{2m+1}| - |X_{2m+1}|$ and $|Y'_{2m+1}| - |Y_{2m+1}|$ are zeros if we perform the SPA steganalysis on the stego image so that the relative error $\frac{||\bigcup_{m=0}^{j} Y_{2m+1}| - |\bigcup_{m=0}^{j} X_{2m+1}||}{|\bigcup_{m=0}^{j} Y_{2m+1}| + |\bigcup_{m=0}^{j} X_{2m+1}|}$ of cover image is unchanged. Under the assumption that $|X_{2m+1}| = |Y_{2m+1}|$, which can be taken for the most natural images, the length of the hidden message that can be estimated from (5) or (6) is zero because $|X_{2m+1}| = |X'_{2m+1}|$ and $|Y_{2m+1}| = |Y'_{2m+1}|$. As a matter of fact, we can directly generate the following equations from (2) and (3) if $|X_{2m+1}| = |X'_{2m+1}|$ and $|Y_{2m+1}| = |Y'_{2m+1}|$:

$$\alpha^2(|C_m| - 4|X'_{2m-1}|) = \alpha(|D'_{2m}| - 2|X'_{2m-1}|), \tag{7}$$

$$\alpha^2(|C_m| - 4|Y'_{2m+1}|) = \alpha(|D'_{2m}| - 2|Y'_{2m+1}|). \tag{8}$$

**Table 1.** The experimental results on distortion and hiding rate

| Images | Size | LSB$^+$: $\theta = 1$ | | LSB$^+$: 4-N, $\theta = 1$ | | |
|---|---|---|---|---|---|---|
| | | PSNR (dB) | Rate (bpp) | PSNR | Bits | Rate |
| airfield | 512×512 | 53.9397 | 0.5064 | 75.1562 | 574 | 0.0021 |
| boats | 720×576 | 51.1656 | 0.9628 | 60.1683 | 33264 | 0.0802 |
| columbia | 480×480 | 51.1480 | 0.9660 | 61.8852 | 12591 | 0.0546 |
| crowd | 512×512 | 51.9641 | 0.6430 | 62.4494 | 12474 | 0.0476 |
| lena | 512×512 | 51.1466 | 0.9688 | 65.1796 | 5564 | 0.0212 |
| lighthouse | 512×512 | 51.3676 | 0.8056 | 67.1048 | 3746 | 0.0143 |
| peppers | 512×512 | 51.1552 | 0.9641 | 67.9355 | 2857 | 0.0109 |
| tank | 512×512 | 57.7708 | 0.2045 | 74.1208 | 668 | 0.0025 |
| truck | 512×512 | 54.4157 | 0.4428 | 66.8681 | 4379 | 0.0167 |

One root of (7) and (8) is zero, and the other root is

$$\alpha = \frac{|D'_{2m}| - 2|X'_{2m-1}|}{|C_m| - 4|X'_{2m-1}|} = \frac{|D'_{2m}| - 2|Y'_{2m+1}|}{|C_m| - 4|Y'_{2m+1}|}, \tag{9}$$

which implies that $(|C_m| - 2|D'_{2m}|)(|Y'_{2m+1}| - |X'_{2m-1}|) = 0$. Because $|X_{2m-1}|$ is unequal to $|Y_{2m+1}|$ for the cover image, we can conclude from (9) that $|C_m| = 2|D'_{2m}|$. Combined with $|C_m| = |X'_{2m-1}| + |D'_{2m}| + |Y'_{2m+1}|$, it can be seen that $|D'_{2m}| = |X'_{2m-1}| + |Y'_{2m+1}|$, which indicates that $\alpha = \frac{1}{2}$. Similarly, we can generate the following equation from (4) given that $|Y_1| = |Y'_1|$:

$$\alpha^2(|C_0| - 2|Y'_1|) = \alpha(|D'_0| - |Y'_1|). \tag{10}$$

Since $|C_0| = |D'_0| + |Y'_1|$ and $|D'_0| \neq |Y'_1|$, the two roots of (10) are 0 and 1. As the value of $\alpha$ (i.e. the fraction of the pixels modified in the embedding process) is zero, the length of the hidden message that is estimated by the SPA steganalysis is also zero. Whether $|X_{2m+1}| = |Y_{2m+1}|$ for $m = 0, 1, \ldots, 127$ or not, the image histogram as well as the values of $|X_{2m-1}|$, $|Y_{2m+1}|$ and $|C_m|$ can be preserved by implementing the LSB$^+$ algorithm on every subset of pixels having the same neighbor values. As a result, the SPA steganalysis is prevented.

## 4 Evaluation

In the experiments, the LSB$^+$ algorithm was implemented on the gray-scale images [1] listed in Table 1 with the parameter $\theta = 1$. In Table 1, we list the data hiding rates when the LSB$^+$ algorithm was implemented on all pixels in an image and on subsets of pixels having the same up, down, left, and right neighbor values (as denoted by 4-N), respectively.

[1] The images are downloaded from http://www.hlevkin.com/TestImages/

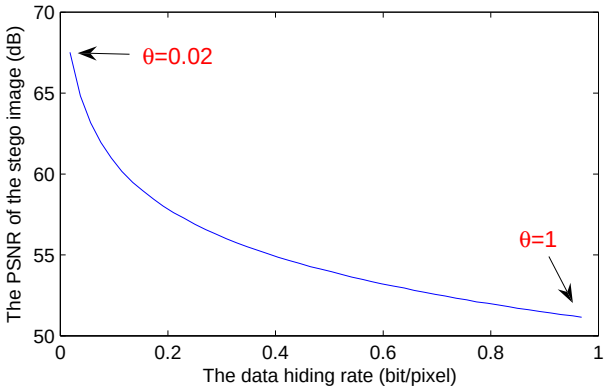



**Fig. 7.** The PSNR of the stego image "lena" at different hiding rate.

### 4.1 Distortion

The peak signal-to-noise ratio (PSNR) of the stego image is used to represent the distortion caused by data hiding. As shown in Table 1, the PSNRs of the stego images are all above 51(dB) when the LSB$^+$ algorithm was implemented on all pixels in a gray-scale image with $\theta = 1$. When the LSB$^+$ algorithm was implemented only on every subset of pixels surrounded by the same neighbor values, the PSNRs of all the stego images were above 60(dB).

### 4.2 Hiding Rate

The information hiding rate using the generic method depends on both of the marginal distribution of the cover and the bin size $\Delta$. In the LSB$^+$ algorithm where $\Delta$ is fixed at 1, the data hiding rate lies on the histogram of the cover image. As we can see in Table 1, less information can be hidden in the image "tank" (about 0.2045 bit/pixel if applied to all pixels) than in the image "lena" (about 0.9688 bit/pixel if applied to all pixels). This is due to that the histogram of "lena" changes slowly while the histogram of "tank" fluctuates rapidly.

Not surprisingly, the distortion of the stego image "tank" is less than that of the stego image "lena". Moreover, we can use the parameter $\theta$ to adjust the data hiding rate so as to tune the perceptual distortion caused by data embedding. As shown in Fig. 7, there is a trade-off between the distortion and the data hiding rate for the stego image of "lena". When implemented on subsets of pixels having the same neighbor values in the 4-N way, the amount of bits that can be embedded is affected by the histogram of the pixels in every subset. It can be seen from Table 1 that the hiding rate has been significantly reduced after restricting the embedding positions to prevent the SPA attack.

### 4.3 The Prevented Steganalytic Algorithms

The LSB$^+$ algorithm is consistent with the model-based steganography, in which two distinct parts are separated from the cover with one part unperturbed and the other replaced with the encoded message. Different from the algorithm of generating the encoded message following a given distribution as in [24], we directly use the cover histogram to generate the stego image so that the hidden message cannot be detected by using the first-order statistics (e.g. [7, 25]). Since the LSB$^+$ algorithm is performed in the spatial domain without differentiating the pixels at the block boundaries and those within the blocks, the steganalysis designed for a block structure (e.g. [5], [9]) or a specific transform domain (e.g. [8], [10]) cannot detect the hidden message either. To further prevent the SPA attack [11], we implement it on the selected subsets of pixels having the same neighbor values. The experimental results show that some important high-order statistics have been well preserved.

### 4.4 Other Steganalysis Using the High-order Statistics

How to prevent the other steganalytic algorithms using the high-order statistics (e.g. [6], [13]-[16]) from detecting the message hidden by the LSB$^+$ algorithm should be further investigated. In principle, it is possible to evade the two attacks against the LSB matching steganography as shown in [13]. The first algorithm calculates the histogram characteristic function (HCF) to calibrate the suspected image with the one down-sampled from it. In the second algorithm, the adjacency histogram is used for steganalysis instead of the usual one. By implementing the LSB$^+$ algorithm on every subset of pixels having the same neighbor values, both the usual and adjacency histograms of the cover image can be preserved. Therefore, the inequality relation between the center of mass (COM) of the stego HCF before and after down-sampling is probably broken. The experimental results on large image database are expected to justify our arguments.

## 5 Conclusion

In this paper, a new steganographic method has been presented for the passive warden scenario. By bijectively mapping the data elements within two adjacent bins to embed a secret message, the first-order statistics of the cover has been preserved inherently. Compared with the previous work in the domain, our method is relative simple and easy to implement. Furthermore, high and adjustable hiding rate can be achieved while the distortion (e.g. in the MSE criterion) can be easily minimized.

The generic method becomes a sort of LSB hiding when applied to digital gray-scale images, namely the LSB$^+$ algorithm. The SPA steganalysis [11] has been prevented by implementing the LSB$^+$ algorithm on the subsets of pixels having the same neighbor values. As a cost, the hiding rate has been significantly reduced by restricting the embedding operations to the selected positions. Our

future work is to investigate how to preserve the high-order statistics so as to prevent the steganalytic attacks as shown in [6], [13]-[16]. We will also try to apply the generic method to some other covers such as 3D objects.

## Acknowledgement

The authors would like to thank the anonymous reviewers for their valuable suggestions and comments. This work was partly supported by the Faculty Research Grant of Hong Kong Baptist University under Project FRG/06-07/II-07 and by a grant from the Research Grant Council of the Hong Kong SAR, China (Project No. HKBU 210306).

## References

1. R. J. Anderson and F. A. P. Petitcolas, "On the limits of steganography," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 474-481, May 1998.
2. G. J. Simmons, "The prisoner's problem and the subliminal channel," *Advances in Cryptology: Proceedings of CRYPTO'83, Plenum Press*, pp. 51-67, 1984.
3. C. Cachin, "An information theoretic model for steganography," *LNCS: Proceedings of the 2nd International Workshop on Information Hiding*, vol. 1525, pp. 306-318, 1998.
4. N. F. Johnson and S. Jajodia, "Steganalysis of images created using current steganography software," *LNCS: Proceedings of the 2nd International Information Hiding Workshop*, vol. 1525, pp. 273-289, 1998.
5. Y. Wang and P. Moulin, "Steganalysis of block-structured stegotext," *Proceedings of the SPIE Electronic Imaging*, vol. 5306, pp. 477-488, San Jose, CA, Jan. 2004.
6. K. Sullivan, U. Madhow, B. S. Manjunath, and S. Chandrasekaran, "Steganalysis for Markov Cover Data with Applications to Images," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 275-287, June 2006.
7. A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems," *LNCS: Proceedings of the 3rd International Workshop on Information Hiding*, vol. 1768, pp. 61-76, 1999.
8. J. Fridrich, M. Goljan, and D. Hogea, "Steganalysis of JPEG images: Breaking the F5 algorithm," *LNCS: Proceedings of the 5th International Workshop on Information Hiding*, vol. 2578, pp. 310-323, 2002.
9. J. Fridrich, M. Goljan, and D. Hogea, "Attacking the OutGuess," *Proceedings of the ACM Workshop on Multimedia and Security*, pp. 967-982, Juan-Pins, France, December 2002.
10. R. Bohme and A. Westfeld, "Exploiting preserved statistics for steganalysis," *LNCS: Proceedings of the 6th International Workshop on Information Hiding*, vol. 3200, pp. 82-96, May, 2004.
11. S. Dumitrescu, X. Wu, and Z. Wang, "Detection of LSB steganography via sample pair analysis," *IEEE Transactions on Signal Processingy*, vol. 51, no. 7, pp. 1995-2007, July 2003.
12. S. Lyu and H. Farid, "Steganalysis using color wavelet statistics and one-class support vector machines," *Proceedings of the SPIE Electronic Imaging*, vol. 5306, pp. 35-45, San Jose, CA, January 2004.

13. A. D. Ker, "Steganalysis of LSB matching in grayscale images," *IEEE Signal Processing Letters*, vol. 12, no. 6, pp. 441-444, June 2005.
14. I. Avcibas, N. Memon, and B. Sankur, "Steganalysis using image quality metrics," *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 221-229, February 2003.
15. S. Lyu and H. Farid, "Steganalysis using high-order image statistics," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 1, pp. 111-119, March 2006.
16. Y. Wang and P. Moulin, "Optimized feature extraction for learning-based image steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 1, pp. 31-45, March 2007.
17. A. Westfeld, "High capacity despite better steganalysis (F5 - a steganographic algorithm)," *LNCS: Proceedings of the 4th International Workshop on Information Hiding*, vol. 2137, pp. 289-302, April 2001.
18. N. Provos, "Defending against statistical steganalysis," *Proceedings of the 10th USENIX Security Symposium*, pp. 323-335, Washington DC, 2001.
19. E. Franz, "Steganography preserving statistical properties," *LNCS: Proceedings of the 5th International Workshop on Information Hiding*, vol. 2578, pp. 278-294, October, 2002.
20. J. J. Eggers, R. Bauml, and B. Girod, "A communications approach to image steganography," *Proceedings of the SPIE Electronic Imaging*, vol. 4675, pp. 26-37, San Jose, CA, 2002.
21. R. Tzschoppe, R. Bauml, J. B. Huber, and A. Kaup, "Steganographic system based on higher-order statistics," *Proceedings of the SPIE Electronic Imaging*, vol. 5020, pp. 156-166, San Jose, CA, Jan 2003.
22. K. Solanki, K. Sullivan, U. Madhow, B. S. Manjunath and S. Chandrasekaran, "Provably secure steganography: Achieving zero K-L divergence using statistical restoration," *IEEE International Conference on Image Processing 2006*, pp. 125-128, Atlanta, USA, Octobor 2006.
23. A. Sarkar, K. Solanki, U. Madhow, S. Chandrasekaran and B. S. Manjunath, "Secure steganography: Statistical restoration of the second order dependencies for improved security," *Proceedings of the 32th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Honolulu, Hawaii, April 2007.
24. P. Sallee, "Model-based Steganography," *LNCS: Proceedings of the International Workshop on Digital Watermarking 2003*, vol. 2939, pp. 154-167, Oct. 2003.
25. R. Bohme and A. Westfeld, "Breaking Cauchy Model-based JPEG Steganography with First Order Statistics," *LNCS: Proceedings of ESORICS 2004, P. Samarati et al (Eds.)*, vol. 3193, pp. 125-140, 2004.