Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®
Pattern Recognition Letters 25 (2004) 331–339
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security

Xinpeng Zhang *, Shuozhong Wang

School of Communication and Information Engineering, Shanghai University, 149 Yanchang Road, Shanghai 200072 PR China

Received 23 June 2003; received in revised form 13 October 2003

Abstract

The pixel-value differencing (PVD) steganography can embed a large amount of secret bits into a still image with high imperceptibility as it makes use of the characteristics of human vision sensitivity. However, a loophole exists in the PVD method. Unusual steps in the histogram of pixel differences reveal the presence of a secret message. An analyst can even estimate the length of hidden bits from the histogram. To enhance security, a modified scheme is proposed which avoids occurrence of the above-mentioned steps in the pixel difference histogram while preserving the advantage of low visual distortion of the PVD. The histogram-based steganalysis is therefore defeated.

Keywords: Steganography; Steganalysis; Pixel-value differencing; Histogram analysis

1. Introduction

Digital watermarking and steganography are two important branches of information hiding (Petitcolas et al., 1999). While watermarking aims to protect intellectual property rights of multimedia contents, the purpose of steganography is to send secrete messages under the cover of a carrier signal. It is generally accepted that a steganographic technique must possess two important properties: good imperceptibility and sufficient data capacity. The first property ensures that the embedded messages are undetectable, and the second means efficiency in hidden communication. Despite that steganographic techniques only alter the most insignificant components, they inevitably leave detectable traces so that successful attacks are often possible. The primary goal of attack on steganographic systems, termed steganalysis, is to detect the presence of hidden data (Wang and Wang, in press), and many steganalytic techniques have been developed (Fridrich and Goljan, 2002).

LSB steganography has low computation complexity and high embedding capacity, in which a secret binary sequence is used to replace the least significant bits of the host medium (Bender et al., 1996). However many steganalytic approaches have been developed to successfully attack the LSB techniques. The χ^2 method (Westfield and Pfitzmann, 1999) detects the presence of hidden data based on the fact that the occurrence probabilities

* Corresponding author. Tel.: +86-2156331435; fax: +86-2156331964.
E-mail address: zhangxinpeng@263.net (X. Zhang).

0167-8655/$ - see front matter
doi:10.1016/j.patrec.2003.10.014

of adjacent gray values tend to become equal after the LSB embedding. The technique can also be used against other steganographic schemes such as J-Steg in which pairs of values (PoVs) are swapped into each other to embed message bits. Another powerful method, RS steganalysis (Fridrich and Goljan, 2002; Fridrich et al., 2001) proposed by Fridrich et al., utilizes a pair of mutually complementary flipping, F_1 and F_{-1}, to test the received image. If the change of smoothness is asymmetric, the image is judged as containing secret message.

An effective way of finding clues of hidden information in still images is to analyze histograms of data samples or other derived parameters in the suspected media. For example, it has been shown that steganographic embedding is equivalent to low-pass filtering the histogram, characterized by a decrease in the mass center of a characteristic function of the histogram, and this decrease can be exploited to identify stego-media (Harmsen and Pearlman, 2003). Another work suggests that, after estimating the original histogram of DCT coefficients, histogram distortion caused by data embedding in JPEG images can be utilized in the steganalysis (Fridrich et al., 2002). Therefore it is important for any steganogrphic technique to withstand histogram-based analyses. To achieve this, a method termed least histogram abnormality (LHA) steganography has been proposed in which the asymmetry inherent in conventional LSB embedding techniques is avoided and the embedding-induced abnormality in the image histogram kept minimum so that the method is immune not only to the χ^2 and RS attacks but also to histogram-based analyses (Zhang et al., 2003).

The pixel-value differencing (PVD) steganography (Wu and Tsai, 2003) proposed by Wu and Tsai can hide a large amount of secret bits into a still image by modifying the difference values between pairs of adjacent pixels. In this technique, more data are inserted into areas where undulation of pixel-values is large as pixels in these areas can tolerate more changes. This leads to good imperceptibility with a high embedding rate.

It is shown in the present paper that, although the PVD steganography can resist the RS analysis as demonstrated in (Wu and Tsai, 2003), it is vulnerable to steganalysis based on the histogram of pixel-value differences. The histogram-based analysis can even provide an estimate of the embedded data length. To enhance security, a modified scheme is proposed. In Section 2 the pixel-value differencing steganography is briefly reviewed. Section 3 describes the histogram-based analysis. Section 4 presents the modification to the PVD method with experimental verification. Section 5 concludes the paper.

2. Pixel-value differencing steganography

In the pixel-value differencing steganography (Wu and Tsai, 2003), a cover image is first segmented into many non-overlapping blocks of two neighboring pixels. The image partitioning may be done by running through all rows in a zigzag manner. A difference d is calculated between the two pixels in each block, $d = p_{i+1} - p_i$, with $|d| \in [0, 255]$. Classify $|d|$ into a number of contiguous ranges, $\boldsymbol{R}_k$ $(k = 0, 1, \ldots, K-1)$ where the width of $\boldsymbol{R}_k$ is a power of 2. A practical set of ranges may be [0 7], [8 15], [16 31], [32 63], [64 127], and [128 255]. The lower bound, upper bound, and width of $\boldsymbol{R}_k$ are denoted l_k, u_k, and w_k respectively. If $|d|$ is in $\boldsymbol{R}_k$, a total of $\log_2(w_k)$ secret bits are embedded into the corresponding 2-pixel block.

Convert the $\log_2(w_k)$ secret bits into a decimal value b, and calculate

$$d' = \begin{cases} l_k + b, & \text{if } d \geqslant 0 \\ -(l_k + b), & \text{if } d < 0 \end{cases} \tag{1}$$

Note that both $|d'|$ and $|d|$ belong to the same $\boldsymbol{R}_k$. The embedding procedure is described as

$$\begin{aligned}(p_i', p_{i+1}') &= f[(p_i, p_{i+1}), d'] \\ &= \begin{cases} (p_i - r_c, p_{i+1} + r_f), & \text{if } d \text{ is odd} \\ (p_i - r_f, p_{i+1} + r_c), & \text{if } d \text{ is even} \end{cases}\end{aligned} \tag{2}$$

where

$$r_c = \left\lceil \frac{d' - d}{2} \right\rceil, \qquad r_f = \left\lfloor \frac{d' - d}{2} \right\rfloor \tag{3}$$

For any block, if there is possibility to cause an overflow by embedding, i.e., to make the value of p_i' or p_{i+1}' out of the range [0 255], the block is

labeled as *unusable* and excluded in the actual embedding process. In general, only a small portion of the blocks is *unusable*. In this way, the greater the difference between adjacent pixels, the more the secret bits are embedded. Therefore busy areas carry more data than smooth areas, achieving large capacity with high invisibility.

In extraction, *usable* and *unusable* blocks can be identified using the same criterion as in the embedding. The value $d' = p'_{i+1} - p'_i$ is obtained in each *usable* block. If $|d'| \in \boldsymbol{R}_k$, the embedded bits are extracted by computing $b = |d'| - l_k$.

3. Steganalysis against PVD embedding

Let the histogram of d, which is the difference between two pixels in a block, be $h(d)$ $(-255 \leqslant d \leqslant 255)$. Generally speaking, in a normal image, the number of occurrences of the pixel difference, $h(d)$, decreases with increasing $|d|$ in a macroscopically smooth fashion. The histogram of pixel differences for a test image Baboon sized 512×512 is shown in Fig. 1.

The effect of PVD embedding on the histogram of pixel differences is analyzed as follows. The data to be hidden can be viewed as a random bit stream since they are usually encrypted before embedding. Thus b is distributed uniformly in $[0,\ w_k - 1]$. Assuming $k > 0$, the numbers of d values falling into $[l_k\ u_k]$, $[-u_0\ u_0]$, and $[-u_k\ -u_k]$ are denoted, respectively, r_k, r_0, and r_{-k}. Thus the pixel difference histogram $h'(d)$ of the stego-image can be obtained:

$$h'(0) \approx (1-\alpha)\cdot h(0) + \frac{\alpha \cdot r_0}{w_0} \tag{4}$$

$$h'(d) \approx (1-\alpha)\cdot h(d) + \frac{\alpha}{w_0}\cdot \sum_{j=0}^{u_0} h(j), \quad 0 < d \leqslant u_0 \tag{5}$$

$$h'(d) \approx (1-\alpha)\cdot h(d) + \frac{\alpha}{w_0}\cdot \sum_{j=-u_0}^{-1} h(j), \quad -u_0 \leqslant d < 0 \tag{6}$$

$$h'(d) \approx \begin{cases} (1-\alpha)\cdot h(d) + \dfrac{\alpha \cdot r_k}{w_k}, \\ \quad \text{if } l_k \leqslant d \leqslant u_k, \\ (1-\alpha)\cdot h(d) + \dfrac{\alpha \cdot r_{-k}}{w_k}, \\ \quad \text{if } -u_k \leqslant d \leqslant -l_k, \end{cases} \quad k > 0 \tag{7}$$

where α is the ratio between the number of blocks containing secret bits and the total number of blocks. Eqs. (4)–(6) are derived from the fact that pixel differences in $[-u_0\ u_0]$ may be changed to 0, but if the original d is 0, it can only be changed into $[0\ u_0]$.

Note that there is a gap between $h'(d)$ and $h'(d+1)$ when $|d|$ and $|d+1|$ belong to two different $\boldsymbol{R}_k$s, because the difference between r_k/w_k and r_{k+1}/w_{k+1} is significantly greater than the difference between $h(d)$ and $h(d+1)$. On the other hand, the greater the value of α, the more the function $h'(d)$ between the two steps approaches a horizontal line. For example, pixel difference histograms of the stego-Baboon containing embedded data in all usable blocks and in 50% of the blocks, respectively, are shown in Fig. 2(a) and (b), with the six ranges of $|d|$ being [0 7], [8 15], [16 31], [32 63], [64 127], and [128 255]. The steps at [7 8], [15 16], [31 32], [63 64], and [127 128] in the pixel difference histograms clearly reveal the presence of secret data. In addition, by carrying out a Fourier analysis on the PVD histogram, the excessive high-frequency components associated with the steps also provide a signature of the secret message, as shown in Fig. 3 by the DFT generated from

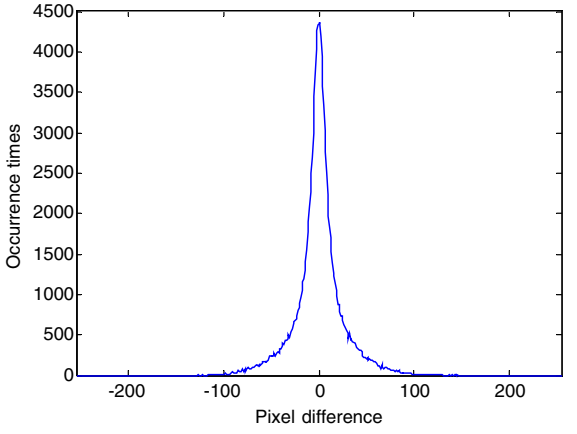

Fig. 1. Histogram of pixel differences, $h(d)$, of the test image Baboon.

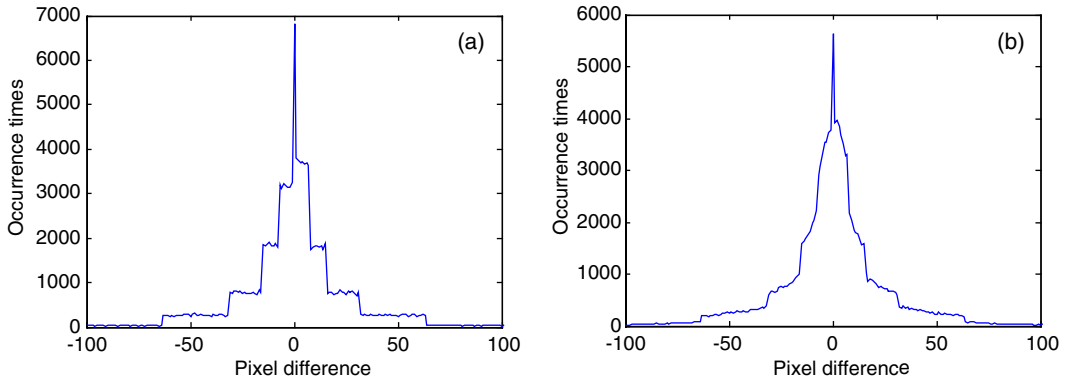

Fig. 2. Pixel difference histogram of stego-image Baboon: (a) $\alpha = 99.6\%$ and (b) $\alpha = 50\%$.

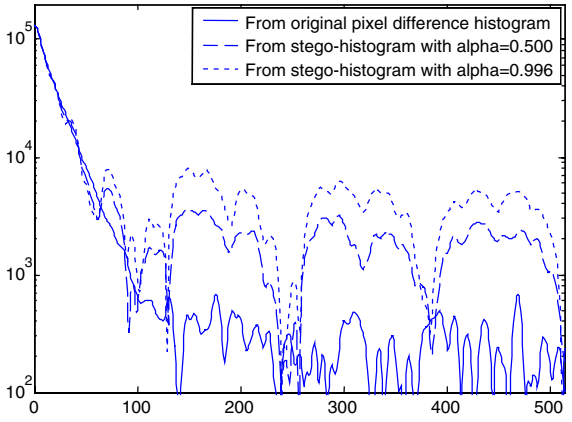

Fig. 3. Spectra generated from PVD histograms of the original and stego images.

histograms of the original and stego images respectively.

Furthermore, the length of hidden data can be estimated from the PVD histogram of the received image, $h'(d)$. By identifying the difference between two successive values in $h'(d)$ which is significantly larger than that between other adjacent values, a steganalyst can reliably find all l_k, u_k, and w_k ($k = 0, 1, \ldots, K-1$). Since data hiding does not change the value of r_k,

$$r_k = \begin{cases} \sum_{d=l_k}^{u_k} h'(d), & k > 0 \\ \sum_{d=-u_0}^{u_0} h'(d), & k = 0 \\ \sum_{d=-u_k}^{-l_k} h'(d), & k < 0 \end{cases} \tag{8}$$

Having obtained all l_k, u_k, and w_k, an estimated histogram of the original PVD, $h_0(d)$, can be worked out from the set of points $[(l_k + u_k)/2,\ r_k/w_k]$ and $[-(l_k + u_k)/2,\ r_{-k}/w_k]$ $(k = 1, \ldots, K-1)$ using spline fitting. On the other hand, $h'(d)$ is smoothed to produce $h'_S(d)$ in each area of $[l_k,\ u_k]$ or $[-u_k,\ -l_k]$ removing small fluctuations while reserving the trend of $h'(d)$. The obtained $h'_S(d)$ will be used in estimating the length of hidden data as it causes less error than $h'(d)$. If the embedding rate α is close to 1, segments in $h'(d)$ within $[l_k,\ u_k]$ or $[-u_k,\ -l_k]$ are almost horizontal. In general, $h'(d)$ is more leveled than $h_0(d)$ in $[l_k,\ u_k]$ or $[-u_k,\ -l_k]$. So, the embedding rate in the kth range can be estimated from the ratio between the slopes of $h'_S(d)$ and $h_0(d)$:

$$\alpha_k = \begin{cases} 1 - \dfrac{\sum_{d=l_k}^{u_k} |h'_S(d) - r_k/w_k|}{\sum_{d=l_k}^{u_k} |h'_0(d) - r_k/w_k|}, & 1 \leqslant k \leqslant K-1 \\ 1 - \dfrac{\sum_{d=-u_{-k}}^{-l_k} |h'_S(d) - r_k/w_{-k}|}{\sum_{d=-u_{-k}}^{-l_k} |h'_0(d) - r_k/w_{-k}|}, & 1-K \leqslant k \leqslant -1 \end{cases} \tag{9}$$

Because a pixel difference in $[-u_0\ u_0]$ can be blank;changed to 0 and the same parameter with an original value 0 can only be changed into $[0\ u_0]$, a special method is used to estimate α_0. Express $h'(0)$ as a sum of two parts: $h'_+(0)$ for the original d in the range of $[0\ u_0]$, and $h'_-(0)$ for the original d in $[-u_0\ -1]$. The former can be estimated by extrapolation from $h'(1)$, $h'(2)$,..., and $h'(u_0)$.

Table 1
Actual length and estimated amount of hidden bits

Cover images	Embedding using the range widths of 8, 8, 16, 32, 64, 128				Embedding using the range widths of 4, 4, 8, 8, 8, 16, 16, 32, 32, 64, 64			
	$\alpha = \alpha_{max}$		$\alpha = 0.500$		$\alpha = \alpha_{max}$		$\alpha = 0.500$	
	L	L_E	L	L_E	L	L_E	L	L_E
Baboon	4.6×10^5	4.4×10^5	2.3×10^5	2.4×10^5	3.6×10^5	3.4×10^5	1.8×10^5	1.6×10^5
Couple	4.1×10^5	4.3×10^5	2.0×10^5	2.2×10^5	3.1×10^5	2.8×10^5	1.5×10^5	1.4×10^5
Lena	4.0×10^5	3.7×10^5	2.0×10^5	2.3×10^5	2.8×10^5	2.5×10^5	1.4×10^5	1.3×10^5
Crowd	4.1×10^5	4.0×10^5	2.1×10^5	2.4×10^5	3.0×10^5	3.3×10^5	1.5×10^5	1.7×10^5

Because of the data embedding, the original d in $[-u_0\ -1]$ is changed to 0 with a probability α_0/w_0. Therefore

$$\alpha_0 = \frac{w_0 \cdot h'_-(0)}{\sum_{j=-u_0}^{-1} h'(j) + h'_-(0)} = \frac{w_0 \cdot [h'(0) - h'_+(0)]}{\sum_{j=-u_0}^{-1} h'(j) + h'_-(0)} \quad (10)$$

Finally, the estimated length of embedded data is obtained:

$$L_E = \sum_{k=1-K}^{K-1} \alpha_k \cdot r_k \cdot \log_2(w_{|k|}) \quad (11)$$

In an experiment, four test images all sized 512×512 were used as the cover media. For cases of $\alpha = \alpha_{max}$ meaning that all *usable* blocks were used for data embedding, and $\alpha = 50\%$, respectively, the actual lengths of hidden bit-sequences L and the estimated lengths L_E are listed in Table 1. Two different ways of dividing the ranges of $|d|$ were used. It is seen that the estimation was quite accurate.

It is obvious that there are many ways of segmenting a cover image into non-overlapping blocks of two neighboring pixels. If the pattern of segmentation for data hiding is determined by a pseudo-random walking based on an encryption key instead of an orderly scanning, the steganalyst can no longer easily obtain $h'(d)$ in the way as described in the above. Nonetheless, by considering the differences between each pixel and all its four adjacent pixels, one can also construct a histogram of pixel differences, referred to as an expanded histogram in the following. If a received image is clean, the histogram should be smooth without pronounced steps. On the other hand, if the received image contains embedded data, steps will occur in the expanded histogram since any pixel forms a block with one of its four neighbors.

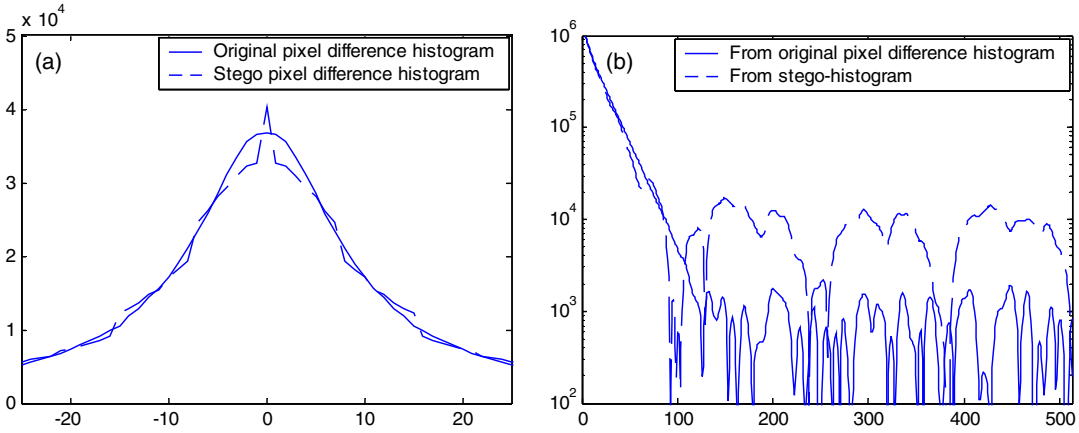

Fig. 4. Analysis of PVD steganogrphy using pseudo-random block-definition: (a) four-neighbor based PVD histograms of original and stego Baboons and (b) spectra of the histograms.

Therefore the vulnerability of PVD steganography is still there. Fig. 4 compares the four-neighbor based pixel difference histograms of the original and stego Baboons with $\alpha = 0.996$ and with the range widths of $|d|$ being 8, 8, 16, 32, 64, and 128. The steps in the PVD histogram and the raised higher frequency components in its spectrum announce the presence of hidden data.

4. Modification to the PVD steganography

It has been shown in the above that the key for a steganalyst to detect the presence of secret messages is the existence of abnormal steps in the pixel difference histogram introduced by data embedding. In order to make the PVD steganography immune to the histogram analysis, measures have to be taken to eliminate the undesired steps. For this purpose, pseudo-randomly select a parameter $\beta \in [0, 1]$, generated from an embedding key, for each block of two consecutive pixels, and calculate

$$\begin{aligned} l'_k &= l_k + \lfloor \beta \cdot w_k \rfloor, \\ u'_k &= l_{k+1} + \lfloor \beta \cdot w_{k+1} \rfloor - 1 \\ &= u_k + \lfloor \beta \cdot w_{k+1} \rfloor = l'_{k+1} - 1 \end{aligned} \tag{12}$$

where k is a range index. Thus, instead of the fixed ranges as used in the original PVD method, the new ranges are defined by the varied l'_k and u'_k. In other words, the ranges corresponding to different blocks are differently defined according to a secret key. Because $w_k \leqslant w_{k+1}$

$$\begin{aligned} u'_k - l'_k &= l_{k+1} + \lfloor \beta \cdot w_{k+1} \rfloor - 1 - l_k - \lfloor \beta \cdot w_k \rfloor \\ &\geqslant w_k - 1 \end{aligned} \tag{13}$$

Eq. (13) indicates that the width of any varied range is no less than that of the original fixed range. If $l'_k \leqslant |d| \leqslant u'_k$ $(k > 1)$, a total of $\log_2(w_k)$ secret bits are embedded into the corresponding block. Convert the secret bits into a decimal value b, and calculate

$$d' = \begin{cases} \underset{l'_k \leqslant e \leqslant u'_k, \bmod(e, w_k) = b}{\arg\min} (|e - d|), & \text{if } d > 0 \\ -\left[\underset{l'_k \leqslant e \leqslant u'_k, \bmod(e, w_k) = -b}{\arg\min} (|e - d|) \right], & \text{if } d < 0 \end{cases} \tag{14}$$

In other words, d' is the value that is closest to d among all values in the same range having a residue $b \bmod w_k$. If $0 \leqslant |d| \leqslant u'_0$ calculate d' from the decimal value b representing $\log_2(w_0)$ secret bits,

$$d' = \underset{-u'_0 \leqslant e \leqslant u'_0, \bmod(e, w_0) = b}{\arg\min} (|e - d|) \tag{15}$$

Modify the two pixels using Eq. (2). As in the original PVD method, the larger the value of d, the more the secret bits are embedded.

Similarly, overflow caused by data embedding should be avoided. If no overflow occurs when $|d'|$ equals the smallest one amongst all u_ks with a value greater than $|d|$, the block is recognized as *usable*. The modification described in this section is performed only in the *usable* blocks.

On the extraction side, b can be restored simply by

$$b = \begin{cases} \bmod(d', w_0), & \text{if } 0 \leqslant |d'| \leqslant u'_0 \\ \bmod(d', w_k), & \text{if } l'_k \leqslant |d'| \leqslant u'_k \quad (k > 0) \end{cases} \tag{16}$$

Note that if β values in all the blocks are 0, the proposed approach degenerates to the original PVD method, and the steps in pixel difference histogram will reveal the presence of hidden data. Nonetheless occurrence of such a case is highly unlikely. Since the introduction of a pseudo-random parameter β makes l'_k and u'_k varying over different blocks, steps in the PVD histogram disappear as demonstrated in Fig. 5(a) obtained respectively from the original and stego Baboons. The stego-Baboon was generated using the proposed modification of PVD with $\alpha = 0.996$ and the range widths of $|d|$ being 8, 8, 16, 32, 64, and 128. Fig. 5(b) shows the spectra of PVD histograms, in which the suspicious behavior in the high-frequency band of the stego-histogram no longer exists.

Table 2 presents $L_{\max}$ the maximum capacity resulting from embedding in all *usable* blocks, and the peak-signal-to-noise ratios (PSNR), for several stego-images obtained with the original and modified PVD methods respectively. It is observed that, with the proposed modification, PSNR of the stego-images with respect to the originals increase by 1.5–4.7 dB, varying from image to image, rep-

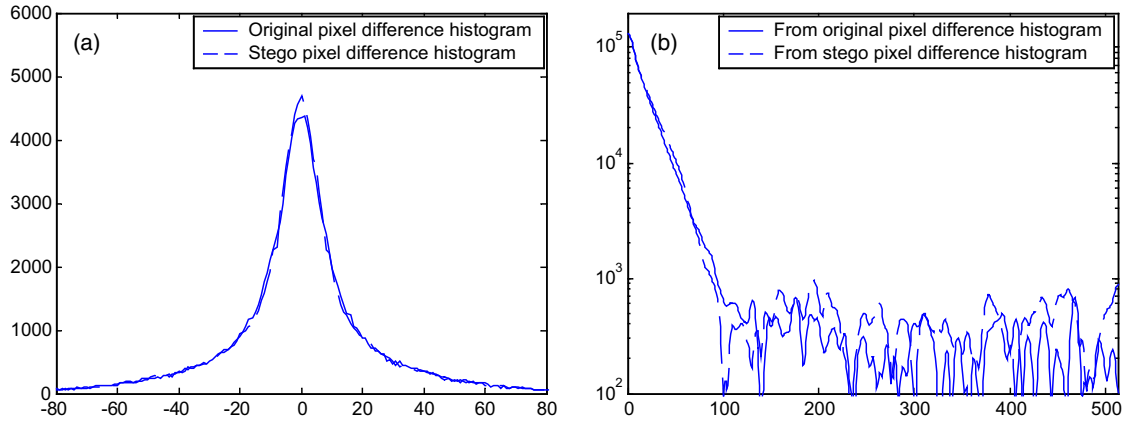

Fig. 5. The modified PVD steganography is immune to histogram analysis: (a) pixel difference histograms of original Baboon and stego-Baboon and (b) spectra of the PVD histograms.

Table 2
Comparison between the original and modified PVD methods

Cover images	Using range widths 8, 8, 16, 32, 64, and 128				Using range widths 4, 4, 8, 8, 8, 16, 16, 32, 32, 64, and 64			
	Original PVD		Modified PVD		Original PVD		Modified PVD	
	L_{max} (bit)	PSNR (dB)	L_{max} (bit)	PSNR (dB)	L_{max} (bit)	PSNR (dB)	L_{max} (bit)	PSNR (dB)
Baboon	4.6×10^5	36.0	4.4×10^5	40.7	3.6×10^5	41.7	3.4×10^4	44.6
Couple	4.1×10^5	40.3	4.0×10^5	43.7	3.1×10^5	45.2	2.9×10^4	47.4
Lena	4.0×10^5	42.7	3.9×10^5	45.1	2.8×10^5	47.7	2.7×10^4	49.2
Crowd	4.1×10^5	40.2	4.0×10^5	43.9	3.0×10^5	45.6	2.9×10^4	47.7

resenting a significant enhancement in imperceptibility. The price paid is only a slight decrease of the embedding capacity. The reason for the resulting improvement is that, in the proposed scheme, values of pixel differences are modified to the nearest one among several candidates. It can also be concluded that if the original image contains more busy areas such as Baboon, more hidden data can be embedded, leading to a lower PSNR. In this case, there is more room for improvement when using the proposed technique.

The RS analysis (Fridrich et al., 2001) was used to test the security of the proposed technique. The RS method defines two mappings: F_1 for $0 \leftrightarrow 1$, $2 \leftrightarrow 3, \ldots, 254 \leftrightarrow 255$ and F_{-1} for $-1 \leftrightarrow 0, 1 \leftrightarrow 2, \ldots, 255 \leftrightarrow 256$. Divide the received image into small blocks of the same size. Define R_M as the ratio of blocks in which the magnitude of fluctuation increases when F_1 is applied to a part of each block, and S_M as the ratio of blocks with decreasing fluctuation magnitudes. In general, $R_M + S_M < 1$. Similarly, another two parameters R_{-M} and S_{-M} are defined when F_{-1} is applied to a part of each block. Statistically, if the image does not contain secret data, F_1 and F_{-1} should equally increase the magnitudes of fluctuation, leading to $R_M \approx R_{-M} > S_M \approx S_{-M}$. When the least significant bits of the cover image are used to carry secret data, the difference between R_M and S_M decreases whereas the difference between R_{-M} and S_{-M} increases because the LSB embedding and the F_1 operation counteract each other. Thus, a comparison between the four parameters can be used to detect the presence of LSB-based embedded data. In the modified PVD steganography, however, the secret data are embedded into the pixel differences instead of the LSBs. The RS analysis should be ineffective. This is verified by

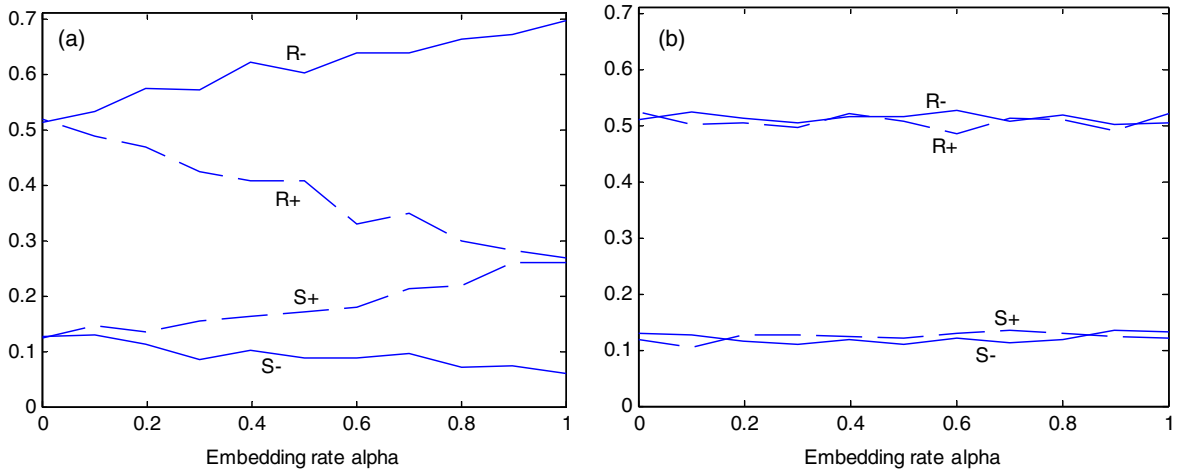

Fig. 6. RS analysis of the stego-Baboon: (a) with the LSB replacement technique and (b) with the modified PVD technique.

experiments. Fig. 6 sketches the RS analysis results for two stego-images using the simple LSB embedding and the proposed method respectively. It is observed that the LSB embedding causes R_M and R_{-M}, as well as S_M and S_{-M}, to separate with an increasing payload, providing a clear signature of the hidden information, whereas the proposed technique is secure under the RS analysis.

5. Conclusion

Since more data are embedded into busy areas than into smooth areas, the pixel-value differencing steganography has a good imperceptibility and considerable embedding capacity. However, the abnormal behavior of the pixel difference histogram reveals the presence of hidden message. After detecting the steps in the histogram, a steganalyst can further estimate the amount of embedded bits. The original PVD method is still vulnerable to the histogram analysis described in this paper even if a pseudo-random pattern is used in defining the pixel doublets.

To enhance security, it is proposed to introduce a pseudo-random dithering to the division of ranges of the pixel-value differences. This effectively removes the undesirable steps existing in the PVD histogram of the stego-image obtained using the original method. In this way, the histogram-based steganalysis is defeated while the advantages of large embedding capacity and high invisibility of the original PVD are preserved.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (60372090).

References

Bender, W., Gruhl, D., Morimoto, N., Lu, A., 1996. Techniques for data hiding. IBM System J. 35 (3–4), 313–336.

Fridrich, J., Goljan, M., 2002. Practical steganalysis of digital images—state of the art. In: Security and Watermarking of Multimedia Contents IV, Proc. SPIE, San Jose, USA, vol. 4675, January 2002, pp. 1–13.

Fridrich, J., Goljan, M., Du, R., 2001. Detecting LSB steganography in color and gray-scale images. Mag. IEEE Multimedia (Special Issue on Security) 1 (October–December), 22–28.

Fridrich, J., Goljan, M., Hogea, D., 2002. Steganalysis of JPEG image: breaking the F5 algorithm. In: 5th Internat. Workshop on Inform. Hiding, Noordwijkerhout, Netherlands, October 2002, pp. 310–323.

Harmsen, J.J., Pearlman, W.A., 2003. Steganalysis of additive noise modelable information hiding. In: SPIE Electronics Imaging, Santa Clara, January 2003. Available from <http://www.cipr.rpi.edu/~harmsj/pubs/harmsen03additive.pdf>.

Petitcolas, F.A.P., Anderson, R.J., Kuhn, M.G., 1999. Information hiding—a survey. Proc. IEEE 87, 1062–1078.

Wang, H., Wang, S., in press. Cyber warfare—steganography vs. steganalysis. Comm. ACM.

Westfeld, A., Pfitzmann, A., 1999. Attacks on steganographic systems. In: 3rd Internat. Workshop on Inform. Hiding Lecture Notes in Computer Science, vol. 1768. Springer, pp. 61–76.

Wu, D.C., Tsai, W.H., 2003. A steganographic method for images by pixel-value differencing. Pattern Recognition Lett. 24, 1613–1626.

Zhang, X., Wang, S., Zhang, K., 2003. Steganography with least histogram abnormality. In: Lecture Notes in Computer Science, vol. 2776, pp. 395–406.