INFORMATICA, 2007, Vol. 18, No. 4, 615–628

Look-up Table Based Reversible Data Hiding for Error Diffused Halftone Images

Jeng-Shyang PAN[1], Hao LUO[1,2], Zhe-Ming LU[3]

[1] *Department of Electronic Engineering, National Kaohsiung University of Applied Sciences*
415 Chien-Kung Road, Kaohsiung 807, Taiwan, ROC
e-mail: jspan@cc.kuas.edu.tw

[2] *Department of Automatic Test and Control, Harbin Institute of Technology*
92 West Dazhi Road, Harbin 150001, P. R. China
e-mail: luohao723@hotmail.com, hluo@bit.kuas.edu.tw

[3] *Visual Information Analysis and Processing Research Center, Harbin Institute of Technology*
Shenzhen Graduate School, Shenzhen Xili Univ. Town, Shenzhen, 518055, P. R. China
e-mail: zhemingl@yahoo.com

Received: April 2006

Abstract. This paper proposes a reversible data hiding method for error diffused halftone images. It employs statistics feature of pixel block patterns to embed data, and utilizes the HVS characteristics to reduce the introduced visual distortion. The watermarked halftone image can be perfectly recovered if it is intact, only a secret key is required. The method is suitable for the applications where the content accuracy of the original halftone image must be guaranteed, and it is easily extended to the field of halftone image authentication.

Key words: look-up table, halftone images, reversible data hiding, human visual system.

1. Introduction

Digital halftoning is a process to transform continuous-tone images into two-tone images, e.g., from 8-bit gray level images to 1-bit binary images. Halftone images can resemble the original versions when viewing from distance by the low-pass filtering of the human eyes. Popular halftoning techniques can be divided into three categories: ordered dithering (Ulichney, 1987), error diffusion (Floyd and Steinberg, 1975) and direct binary search (Lieberman and Allebach, 1996). Among these, error diffusion achieves a preferable tradeoff between the high visual quality and the reasonable computational complexity. With the wide proliferation of error diffused halftone images in books, magazines, printer outputs and fax documents, it is desirable to embed data in this kind of images for copyright protection, content authentication and tamper detection. Different from gray level or colour images, there are mainly three challenges to embed data in halftone images. The first one is less information redundancy for each pixel value is either black or white. Consequently many data hiding approaches such as some transform domain based techniques cannot be directly transplanted to halftone images. Another challenge is visual quality degradation. To insert data in halftones, change of the pixel value is either from black to white or vice versa. According to the study of human visual system (HVS), human eyes are sensitive to the abrupt change aroused by data embedding, e.g., new appearance of the white cross and the black cross. The third one is lower data capacity compared with that of continuous-tone image. High capacity is one of the key factors to evaluate the performance of data hiding techniques. In fact, for halftone images, this challenge is closely related to the former two challenges. It is expected that a large quantity of data is difficult to be embedded into halftone images considering visual quality degradation, for less information redundancy can be employed.

In recent years, several data hiding methods for halftone images, e.g., Tseng and Chang (2005) are reported. These approaches can be divided into three classes: (1) pixel-based: this kind of methods is to change the values of individual pixels usually randomly selected. (2) block-based: these approaches partition the original image into pixel blocks and modify characteristics of some blocks. (3) hybrid-based: they insert data by combining the characteristics of pixel-based and block-based. Nevertheless, most existing data hiding methods cannot recover the original image because of the irreversible distortion introduced. This may not satisfy the requirements of some applications where the content accuracy of the host medium must be guaranteed. Therefore, it is quite necessary to develop a reversible data hiding method for halftone images. However, up to now, there have been few attentions paid to this research theme.

This paper presents a hybrid-based method. The original idea is motivated from the R-S algorithm developed in Fridrich *et al.* (2002). Firstly, a look-up table (LUT) is constructed. It consists of pairs of similar block patterns which are selected according to the statistics combining characteristics of the HVS. We use “0” and “1” to denote the states of two group patterns of the LUT respectively. Secondly, we search all blocks in the image: if one is the same as some pattern in the LUT, record its state. Thus a state sequence can be obtained. Thirdly, this sequence is losslessly compressed and the saved space is filled with the hidden data and some additional information *AI*. Here the *AI* refers to extra data aroused by the LUT embedding. Fourthly, data is hidden by similar patterns toggling with reference to the new sequence. The last step is to insert the LUT with a secret key, and meanwhile the watermarked halftone image is obtained. In data extraction stage, the LUT must be recreated first and other procedures are just the inverse process of data hiding. As a reversible technique, the original image can be perfectly recovered if the watermarked version is intact. Furthermore, our approach is also easily extended for halftone image authentication, e.g., hiding a hash sequence.

The rest of this paper is organized as follows. Section 2 describes the proposed method. In Section 3, experimental results are presented and discussed. Section 4 concludes the whole paper and open perspectives for future work.

2. Proposed Method

2.1. *Pattern Histogram*

The original image is partitioned into a set of non-overlapping 4×4 blocks. This step aims to choose some appropriate blocks to embed data. Obviously a 4×4 block has totally 2^{16} different patterns. In most cases, a majority of patterns never appear or appear only once in an image. We rearrange a 4×4 block into a binary sequence, and transform it into a decimal integer. Therefore, each pattern is uniquely associated with an integer in the range of $[0, 2^{16} - 1]$, which we call a Pattern Index (PI). Each PI is associated with a value named the Number of Appearance Times (NAT) of this pattern in the image. The NAT is counted for each PI and thus the Pattern Histogram (PH) can be constructed. Fig. 1 shows the PHs of six 512×512 halftone images with the x-axis and y-axis denoting the PI and the NAT respectively. It is obvious that a small portion of patterns appear many times, while others appear rarely. This statistics feature can be employed to insert data.

2.2. *Human Visual System Characteristics*

According to the study on the HVS (Pappas and Neuhoff, 1999), the spatial frequency sensitivity of human eyes is usually estimated as a modulation transfer function. Specifically, the impulse response to a printed image of 300 dpi at a viewing distance of 30 inches is virtually identical to that of a Gaussian filter with $\sigma = 1.5$ and $\tau = 0.0095°$. In our research, we adopt the 5×5 visual response filter given in Cheung and Chan (2004) as

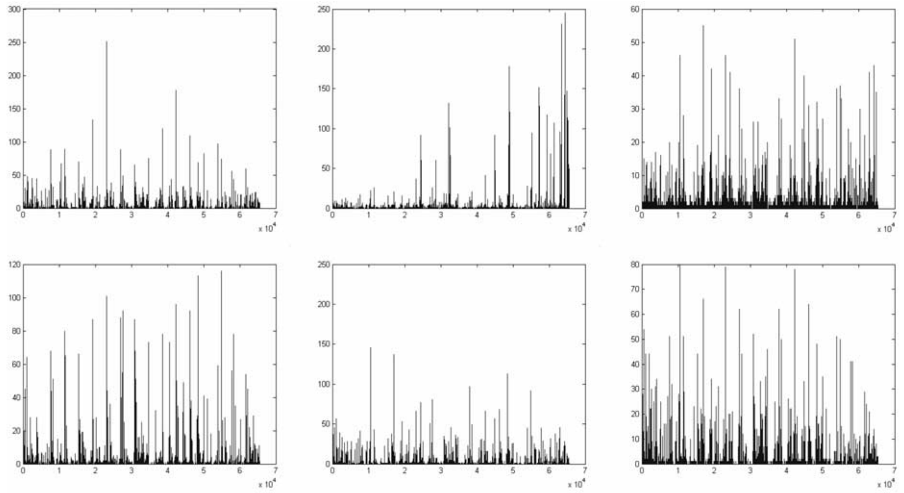


Fig. 1. Pattern histograms of Lena, Airplane, Baboon (above row, from left to right), Boat, Pepper and Barbara (below row, from left to right).

follows

$$f = \frac{1}{11.566}\begin{bmatrix} 0.1628 & 0.3215 & 0.4035 & 0.3215 & 0.1628 \\ 0.3215 & 0.6352 & 0.7970 & 0.6352 & 0.3215 \\ 0.4035 & 0.7970 & 1 & 0.7970 & 0.4035 \\ 0.3215 & 0.6352 & 0.7970 & 0.6352 & 0.3215 \\ 0.1628 & 0.3215 & 0.4035 & 0.3215 & 0.1628 \end{bmatrix}. \tag{1}$$

2.3. *Look-up Table Construction*

Based on the statistics of the PH and HVS characteristics, some patterns are chosen and classified into two groups, H and L. Suppose $H = \{h_1, h_2, \dots, h_u\}$ denotes the group with the first u biggest NATs, and n_i denotes the NAT of Pattern h_i, obviously we have

$$n_k \geqslant n_l, \tag{2}$$

where $1 \leqslant k < l \leqslant u$. The other group $L = \{l_1, l_2, \dots, l_v\}$ is composed of v patterns with NAT $= 1$. Similar pattern pairs $\{h_i, h'_i\}$ $(1 \leqslant i \leqslant u)$ constitute the LUT, where $h'_i \in L$ is the block similar to h_i. Therefore the LUT construction is reduced to the task of finding h'_i in L. In addition, we need to compute the size of the LUT in advance for the hiding capacity is controlled by it. In this research, the size of the LUT is determined by the number of similar pattern pairs I, and it can be transformed into a $32I$ bits binary sequence for each pattern contains 4×4 pixels. Since the LUT is also embedded into the original image, the valid capacity P for hidden data is

$$P = \left(\sum_{i=1}^{I} n_i\right) + I - l_{S_c} - 32I, \tag{3}$$

where $\sum_{i=1}^{I} n_i$ and I are the sums of NATs in the groups H and L, respectively. $32I$ is the size of the *AI* aroused by the LUT embedding. l_{S_c} denotes the length of the compressed state sequence S, the detailed description is shown in the next section.

Suppose D is the length of the hidden data W, the flow chart of the LUT construction is shown in Fig. 2 with steps as follows.

(1) Initialize $i = 1$, $I = 1$, and construct the PH.

(2) Find in L the similar pattern h'_i for h_i, and insert $\{h_i, h'_i\}$ into the LUT. If $P \geqslant D$, go to step (4); otherwise, go to step (3).

(3) $i = i + 1$, $I = I + 1$, and go to (2).

(4) The LUT is attained.

To find h'_i in the step (2), we convolute h_i with f and also all patterns in L with f respectively as follows

$$h_{f_i} = h_i * f, \tag{4}$$

$$l_{f_j} = l_j * f, \tag{5}$$

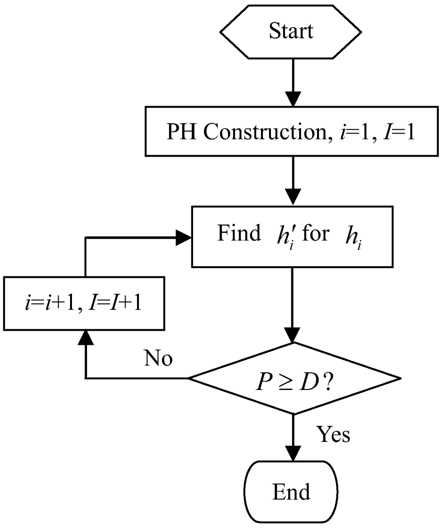


Fig. 2. Flow chart of the LUT construction.

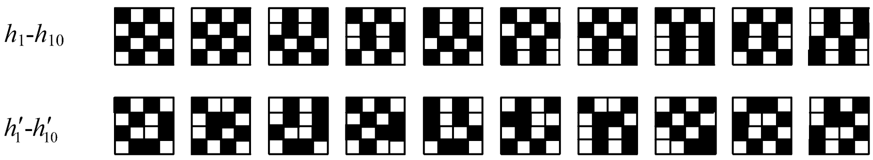


Fig. 3. An example of a look-up table obtained from the Lena image.

where $j = 1, 2, \ldots, v$. Obviously the size of the convolution results h_{f_i} and l_{f_j} is 8×8 (the 4×4 matrix convoluted with the 5×5 matrix). Next, the Euclidean distance d_{ij} computed with Formula (6) is used to measure the similarity between h_{f_i} and l_{f_j}. The l_{f_j} with the smallest d_{ij} from h_{f_i} is recorded and the associated pattern in L is selected as the h_i'.

$$d_{ij} = \sum_{u=1}^{8} \sum_{v=1}^{8} \left[h_{f_i}(u, v) - l_{f_j}(u, v) \right]^2. \tag{6}$$

An example LUT is shown in Fig. 3, which is constructed from the 512×512 halftone Lena image. Only the patterns $h_1 - h_{10}$ (the above row) with the first 10 biggest NATs and their similar patterns $h_1' - h_{10}'$ (the below row) are given.

2.4. *Data Hiding*

The block diagram of the data hiding process is shown in Fig. 4. We use “0” and “1” to denote the states of h_i and h_i'. Once the LUT is created, the process of hiding data is

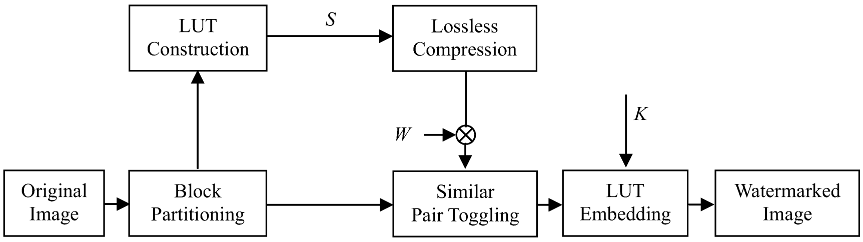


Fig. 4. Block diagram of the data hiding process.

transformed into a simple operation of $h_i - h_i'$ replacement, whose steps can be presented as below.

(1) Partition the original image into non-overlapping 4×4 blocks.

(2) Construct a LUT based on the PH.

(3) Search all blocks in the original image. As long as we come across a pattern in the LUT, we record its state based on the following rule: if it belongs to H, "0" is recorded; otherwise if it belongs to L, then "1" is recorded. Thus the state sequence S can be obtained. In fact, suppose l_{S} denotes the length of S, we have

$$l_S = \left(\sum_{i=1}^{I} n_i \right) + I. \tag{7}$$

(4) Compress S into S_c based on certain lossless entropy coding method, in our case, the arithmetic coding algorithm is used. Here the length of S_c is l_{S_c}.

(5) Noting that the length of the saved space is $l_S - l_{S_c}$. We fill the space $l_S - l_{S_c} - 32I$ and the space $32I$ with the hidden data W and the *AI* respectively, as shown in Fig. 5. Thus we get a new state sequence $S' = \{s_1', s_2', ..., s_{l_s}'\}$. Note that the length of S' is still equal to l_S.

(6) Modulate the states of patterns belonging to the LUT to be the states in S' based on the following rule: If s_k' $(1 \leqslant k \leqslant l_s)$ equals "0" and the current pattern c_p is exactly the same as h_i', then we replace h_i' with h_i; Else if s_k' equals "1" and c_p is exactly the

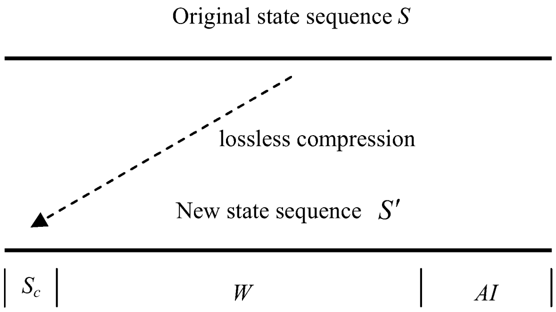


Fig. 5. State sequence lossless compression and the saved space allocation.

same as h_i, then we replace h_i with h_i'. For the other two cases, patterns are unchanged. These operations can be expressed as follows

$$\begin{cases} h_i' \leftarrow h_i & \text{if } s_k' = 0,\ c_p = h_i'; \\ h_i \leftarrow h_i & \text{if } s_k' = 0,\ c_p = h_i; \\ h_i' \leftarrow h_i' & \text{if } s_k' = 1,\ c_p = h_i'; \\ h_i \leftarrow h_i' & \text{if } s_k' = 1,\ c_p = h_i; \end{cases} \tag{8}$$

where ← means replacing the left pattern with the right one. Hence, data is hidden through above similar pair toggling operation.

2.5. *Look-up Table Embedding*

Since different images has different lookup tables, no universal table is suitable for all images. Besides, the lookup table is also need to be protected. Therefore, the LUT is also embedded in the image in this research, with the steps are described as follows.

It is rearranged into a $32I$ bits binary sequence. A secret key K is used to generate $32I$ random pixel locations to embed the LUT, Noting that these pixels must not fall into the blocks in the LUT. We extract the selected pixel values into a $32I$ binary sequence, and embedded it based on Formula (8). Then we directly replace the selected pixels with the LUT sequence.

This priciple is illustrated in Fig. 6, the red blocks denote patterns used to extract state according to the LUT. While the blue points denote the select pixels with K, these pixels values are extracted as *AI*, and then the rearranged LUT (a binary sequence) is inserted into these locations. For example, if the LUT shown in Fig. 3 is to be inserted, we need to select 320 pixel positions (blue points) and replace their pixels values using the rearranged LUT, and the original 320 pixles values are hidden as the way of watermark embedding (red blocks). More details of the LUT embedding method can be seen in Pan *et al.* (2006).

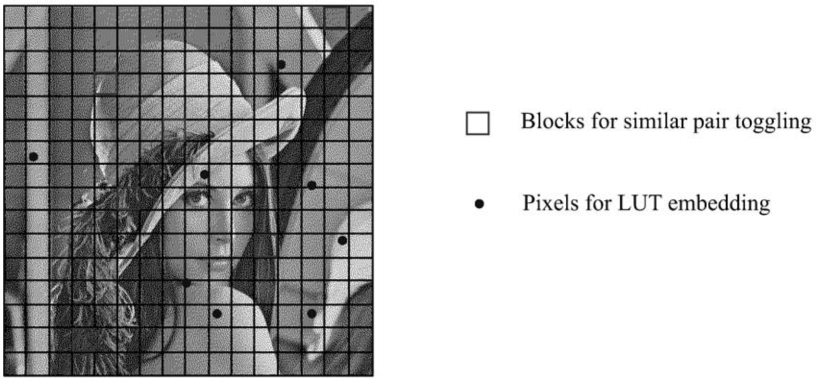

□ Blocks for similar pair toggling
● Pixels for LUT embedding

Fig. 6. Look-up table embedding.

2.6. *Data Extraction*

The block diagram of data extraction is shown in Fig. 7, with steps as follows.

(1) The LUT reconstruction: we use the same Key K to find the $32I$ pixel locations and arrange the pixel values into the LUT.

(2) Extract the state sequence S' according to the LUT.

(3) Lossless decompression: S' can be divided into three parts: the compressed version of S, the hidden data, and the pixel values selected by K. We use the arithmetic decoding method to decode the first part of S' to get S. Then, the hidden data is extracted form the middle $l_S - l_{S_c} - 32I$ bits. The process is illustrated in Fig. 8.

(4) *AI* Recovery: We directly extract the last $32I$ bits of S' to recover the pixel values occupied by the LUT.

(5) Similar pair toggling: Demodulate the states of patterns belonging to the LUT to be the states in S based on the following rule: If S_k $(1 \leqslant k \leqslant l_s)$ equals "0" and the c_p is exactly the same as h'_i, then we replace h'_i with h_i; Else if S_k equals "1" and the c_p is exactly the same as h_i, then we replace h_i with h'_i. For the other two cases, block patterns

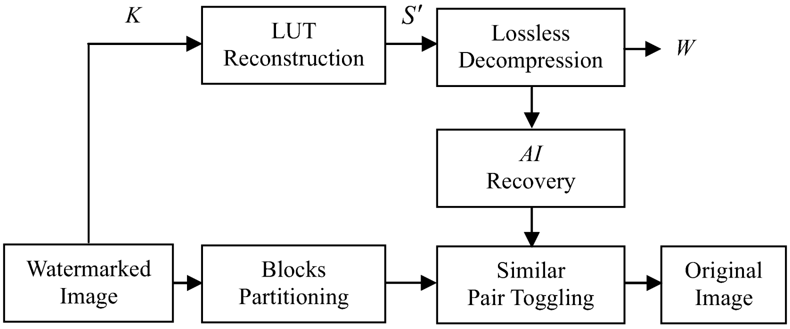


Fig. 7. Block diagram of the data extraction process.

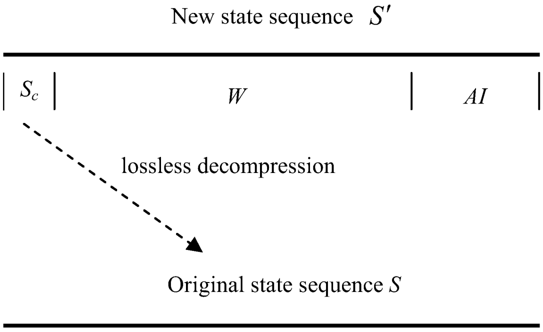


Fig. 8. State sequence lossless decompression for the hidden data extraction and the additional information recovery.

are unchanged. These operations can be described as follows

$$\begin{cases} h_i \leftarrow h'_i & \text{if } s_k = 0, \ c_p = h'_i, \\ h_i \leftarrow h_i & \text{if } s_k = 0, \ c_p = h_i, \\ h'_i \leftarrow h'_i & \text{if } s_k = 1, \ c_p = h'_i, \\ h'_i \leftarrow h_i & \text{if } s_k = 1, \ c_p = h_i. \end{cases} \tag{9}$$

3. Experimental Results

Six 512×512 error diffused halftone images, Lena, Baboon, Airplane, Boat, Pepper and Barbara are selected to test the performance of the proposed method, as shown in Fig. 10. These halftones are obtained by performing Floyd–Steinberg error diffusion filtering on the 8-bit gray level images. Capacities for different images and different sizes of LUT are listed in Table 1.

In our experiments, a 1D binary sequence created by a pseudo-random number generator is chosen as the hidden data. Fig. 11(a) and Fig. 11(b) illustrate the original image Lena and its watermarked version, whereas Fig. 11(c) shows the recovered one. To evaluate the introduced distortion, we apply an effective quality metric proposed by Valliappan

Table 1

Capacity (bits) with different images and different sizes of LUT (I)

LUT	Lena	Airplane	Baboon	Boat	Pepper	Barbara
$I = 1$	201	195	8	68	152	33
$I = 2$	339	385	21	142	258	73
$I = 3$	432	522	28	204	355	112
$I = 4$	512	635	37	261	429	140
$I = 5$	582	742	41	314	487	165
$I = 6$	641	844	45	366	540	188
$I = 7$	690	936	48	416	582	212
$I = 8$	739	1025	51	464	620	228
$I = 9$	788	1109	52	512	655	241
$I = 10$	831	1191	54	553	685	254

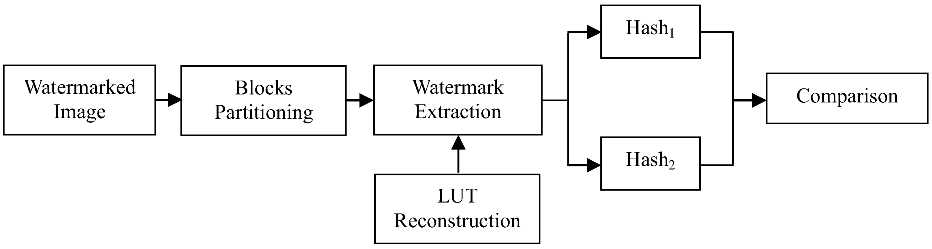


Fig. 9. Application of our scheme in halftone image authentication.

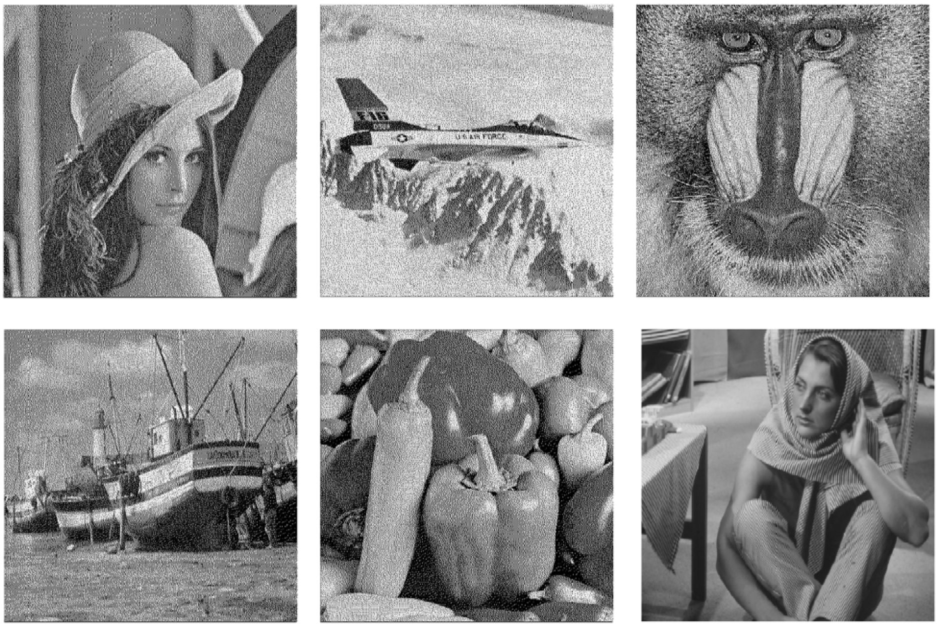
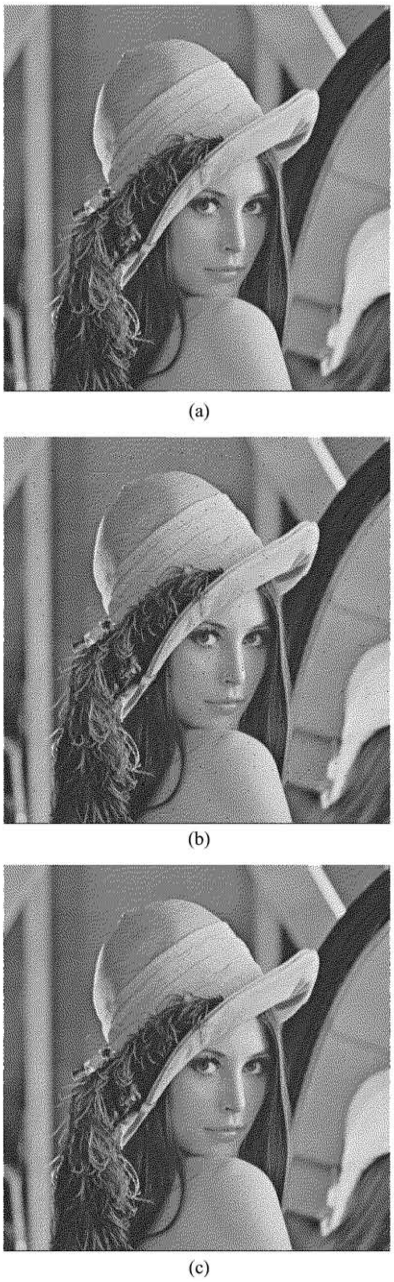


Fig. 10. Six test error diffused images Lena, Airplane, Baboon (above row, from left to right), Boat, Pepper and Barbara (below row, from left to right).

et al. (1999), i.e., weighted SNR (WSNR). The linear distortion is quantified in Valliappan *et al.* (1999) by constructing a minimum mean squared error Weiner filter, in this way the residual image is uncorrelated with the input image. The residual image represents the nonlinear distortion plus additive independent noise. Valliappan *et al.* (1999) spectrally weight the residual by a contrast sensitivity function (CSF) to quantify the effect of nonlinear distortion and noise on quality. A CSF is a linear approximation of the HVS response to a sine wave of a single frequency, and a lowpass CSF assumes the human eyes do not focus on one point but freely moves the eyes around the image.

Since the halftone image is attempted to preserve the useful information of the gray level image, we compare the halftone or watermarked image with the original gray level image. Similar to PSNR, a higher WSNR means higher quality. In our experiments, the WSNR between the gray level Lena and the halftone Lena is 29.18dB, while the WSNR between the gray level Lena and the watermarked Lena is 28.59dB. It can be seen that the introduced distortion of the visual quality is slight. Since the WSNR between the gray level Lena and the recovered Lena is 29.18dB, the recovered version is exactly the same as the original image.

Our method also can be used for halftone image authentication. For example, a hash sequence of the original halftone image can be hidden in the halftone image. We only need to compare the hash extracted from the watermarked image (Hash_1) with the hash sequence computed from the recovered image (Hash_2). When these two sequences are equal, we can confirm that the watermarked image suffers no alteration. Under no attacks,



(a)

(b)

(c)

Fig. 11. Data Hiding on the halftone Lena, (a) the original Lena, WSNR = 29.18dB, (b) the watermarked Lena with 831 bits inserted, WSNR = 28.58dB, (c) the recovered Lena, WSNR = 29.18dB.

both of them are certainly equal to the original hash. While if the watermarked image is unauthorized changed, the two sequences are different. The process is illustrated in Fig. 9.

4. Conclusions

A reversible data hiding method for error diffused halftone images is proposed in this paper. The hidden data can be extracted with a secret key and meanwhile the original image is perfectly recovered. Experimental results show the effectiveness of the method. Since the characteristics of the HVS are exploited, lower degradation is introduced to the visual quality of the image. Our method in this research only focuses on the gray level halftone images, and it is expected to be extended to the colour halftone images in the future.

References

Cheung, S.M., and Y.H. Chan (2004). A technique for lossy compression of error-diffused halftones. In *Proc. of International Conference on Multimedia & Expo*. pp. 1083–1086.

Floyd, R.W., and L. Steinberg (1975). An adaptive algorithm for spatial gray scale. In *Proc. of SID 75 Digest. Society for Information Display*. pp. 36–37.

Fridrich, J., M. Goljan and R. Du (2002). Lossless data embedding – new paradigm in digital watermarking. Special Issue on *Emerging Applications of Multimedia Data Hiding*, **2002**(2), 185–196.

Lieberman, D., and J. Allebach (1996). Digital halftoning using direct binary search. In *Proc. of International Conference on High Technology*. pp. 114–124.

Pan, J.S., H. Luo and Z.M. Lu (2006). A lossless watermarking scheme for halftone image authentication. *International Journal of Computer Science and Network Security*, **6**(2B), 147–151.

Pappas, T.N., and D.L. Neuhoff (1999). Least-squares model based halftoning. *IEEE Transactions on Image Processing*, **8**(8), 1102–1116.

Tseng, H.W., and C.C. Chang (2005). Hiding data in halftone images. *Informatica*, **16**(3), 419–430.

Ulichney, R.A. (1987). *Digital Halftoning*. MIT Press, Cambridge, MA.

Valliappan, M., B.L. Evans, D.A.D. Tompkins and F. Kossentini (1999). Lossy compression of stochastic halftones with JBIG2. In *Proc. of International Conference on Image Processing*. pp. 214–218.

J.-S. Pan received the BS degree in electronic engineering from the National Taiwan University of Science and Technology, Taiwan in 1986, the MS degree in communication engineering from the National Chiao Tung University, Taiwan in 1988, and the PhD degree in electrical engineering from the University of Edinburgh, U.K. in 1996. Currently, he is a professor in the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Taiwan. His current research interests include data mining, information security and image processing.

H. Luo received the BS degree and the MS degree from Harbin Institute of Technology (HIT), Harbin, China in 2002 and 2004, respectively. He is a PhD candidate in the Visual Information Analysis and Processing Research Center, Harbin Institute of Technology Shenzhen Graduate School. His research interests include information security and image processing.

Z.-M. Lu received the BS, MS and PhD. degrees in electrical engineering from Harbin Institute of Technology in 1995, 1997 and 2001, respectively. He was the Alexander von Humboldt research fellow in University of Freiburg in Germany, from Oct., 2004 to Jan. 2006. He is now the professor and director of the Visual Information Analysis and Processing Research Center, Harbin Institute of Technology Shenzhen Graduate School. His research interests are image processing, pattern recognition, information hiding and visual information retrieval.

Paieškos lentele paremtas atstatomas duomenų paslėpimas klaidų išsklaidytuose pustonių paveiksluose

Jeng-Shyang PAN, Hao LUO, Zhe-Ming LU

Straipsnyje pasiūlytas atstatomas duomenų paslėpimo klaidų išsisklaidytuose pustonių paveiksluose metodas. Jame naudojama statistinė pikselių blokų formų savybė įtvirtinti duomenis ir HVS charakteristika sumažinti vaizdo iškraipymus. Nesugadintas vandens ženklais pažymėtas pustonių paveikslas gali būti visiškai atstatytas, bet slaptas raktas yra reikalingas. Metodas yra tinkamas taikymams, kuriuose originalaus pustonio paveikslo turinio tikslumas turi būti garantuotas, be to jis gali būti lengvai pritaikytas pustonio paveikslo autentikavimui.