Interleaving Max-Min Difference Histogram Shifting Data Hiding Method

Hsien-Wei Yang
Overseas Chinese University/Department of Information Management, Taichung, Taiwan
Email: fiber@ocu.edu.tw

Kuo Feng Hwang*
Overseas Chinese University/Department of Information Management, Taichung, Taiwan
Email: kfhwang@ocu.edu.tw

Shou-Shiung Chou
Tunghai University/Department of Hospitality Management, Taichung, Taiwan
Email: shchou@thu.edu.tw

Abstract—**This work proposes a reversible data hiding algorithm that is based on the interleaving max-min difference histogram. The method divides the cover image into non-overlapping identical blocks. In each block, the maximum or minimum pixel is selected to calculate absolute differences between the gray level value of the selected pixel and that of the other pixels. Then, these differences are used to generate a histogram and the histogram shifting method is adopted to embed data. This method, to propose an offset distortion method, exploits the feature that the change of the pixel gray level upon shifting and embedding obtained using the max difference is contrary to that obtained using the min difference. The method of interleaving uses the max difference and the min difference to counteract the change in the gray levels of some pixels. The total of the changed gray level values is thus reduced. Therefore, the PSNR value can be increased and the quality of the stego image improved. Experimental results reveal that the proposed method yields a higher embedding capacity than that of the median difference histogram.**

Index Terms—**reversible data hiding, histogram shifting, difference, max difference, min difference**

I. Introduction

Reversible data hiding [1,2], also known as lossless data hiding, enables marked media to be restored to their original form without any distortion. This technique is applied in such fields as content authentication of multimedia data, law enforcement, medical imagery and astronomical research. Various reversible data hiding methods have been proposed for grayscale images, and these can be divided into the following categories:

1) Data compression

Fridrich *et al.* [3,4] compressed the least significant bit (LSB) plane to obtain additional space for embedding secret data. Celik *et al.* [5-7] improved the method of Fridrich *et al.* and proposed the Generalized-LSB (G-LSB) scheme by compressing the quantization residuals of pixels to yield additional space to embed a message. Awrangjeb and Kankanhalli [8,9] presented a scheme that detects the textured blocks; extracts the LSBs of the pixel-values from these textured blocks based on the Human Visual System (HVS), and concatenates the authentication information with the compressed bit-string.

2) Difference expansion

Tian [10] presented a method that expands the difference between two neighboring pixels to obtain redundant space for embedding a message. Alattar [11] used the difference expansion of vectors of adjacent pixels to obtain more embedding space. Using Tian′s scheme of difference expansion, Chang and Lu [12] calculated the difference between a pixel and the mean value of its neighboring pixels to embed a message. Weng *et al.* [13,14] used the correlations among four pixels in a quad, and embedded data by expanding the differences between one pixel and each of its three neighboring pixels.

3) Histogram shifting

Ni *et al.* [15] used the histogram of the pixel values in the cover image to embed secret data into the maximum frequency pixels. Fallahpour and Sedaaghi [16] divided a cover image into several blocks, and embedded secret data into the histogram of each block. Lin and Hsueh [17] applied the histogram shifting of Ni *et al.* to the pixel differences, which were obtained from the absolute differences between pairs of neighboring pixels.

4) Integer wavelet transform

* Corresponding author. Tel:+886 4 27016855 ext. 1381; Fax:+886 4 27026986

doi:10.4304/jsw.5.6.615-621

Xuan *et al.* [18-20] proposed a lossless data hiding technique based on integer wavelet transform, which embeds high capacity data into the most insensitive bit-planes of wavelet coefficients. Yang *et al.* [21] presented a symmetrical histogram expansion scheme in the transform domain of Piecewise-Linear Haar (PLHaar). Data are embedded into the PLHaar coefficients of the images from the pivotal bin of a histogram of PLHaar coefficients symmetrically toward both sides of the histogram.

The goal of these methods is to realize a high embedding capacity while yet maintaining high image quality. The proposed scheme uses the block difference to generate histogram to embed data, and propose an offset distortion method to reduce the total of the changed gray level value. Experimental results reveal that the proposed method has a greater embedding capacity and produces less image distortion than previous schemes.

The rest of this paper is organized as follows. Section 2 describes the proposed scheme. Section 3 summarizes the experimental results. Section 4 draws conclusions.

II. PROPOSED SCHEME

The proposed interleaving max-min difference histogram method is based on block difference histogram and an offset distortion method (called interleaving max-min difference), and they are described below.

A. Block Difference Histogram

The embedding process (called EP) and extraction process (called E′P) are described below.

1) Embedding Process

The block difference histogram method is shown as Figure 1. The step 1－5 are described below.

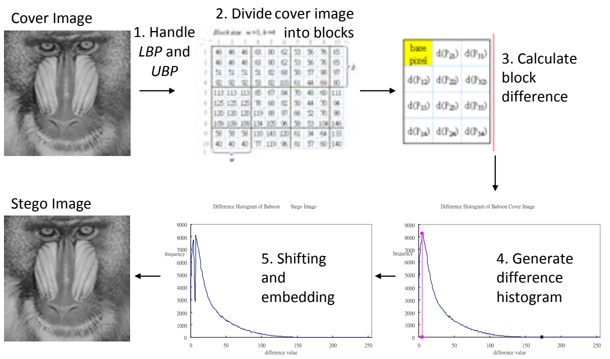

Figure 1. Block difference histogram method

Step 1. Handle *LBP* and *UBP*

In *EP*, pixels with a gray level 0 (called lower bound pixels [*LBP*]) will cause underflow in the reduce operation. Pixels with a gray level 255 (called upper bound pixels [*UBP*]) will cause an overflow in the increase operation. To avoid these problems, the *LBP* and *UBP* are handled before dividing cover image (called *CI*). The gray levels of *LBP* and *UBP* are changed to 1 and 254, respectively. To recover the *LBP* and *UBP* in *E′P*, the information about the *LBP* and *UBP* (called *LUBPI*) is recorded as overhead information (called *OI*).

Step 2. Divide cover image into blocks

The *CI* is divided into non- overlapping identical $w \times h$ blocks, *w* and *h* are the width and height of the block, respectively. Record *w* and *h* as *OI*.

Step 3. Calculate block difference

In a block, select a pixel as the base pixel and each other pixel can get an absolute difference of gray level value (called difference value) between the base pixel and it. The calculated difference values are called block difference.

Step 4. Generate difference histogram

Let $f(x)$ denotes the frequency of a difference value x, where "frequency" is the number of instances of a difference value. Then the difference histogram is $\{ f(x) \mid 0 \leqq x \leqq 255 \}$.

Step 5. Shifting and embedding

Find the difference value of maximum and minimum frequency in the difference histogram, a and b. The area between a+1 and b-1 are shifted to right by one, and the frequency of a+1 is leaved unoccupied. Then the frequency of a is used to embed data, a pixel has difference value a can embed one bit. If the embedded data is bit 1, then the difference value is increased by one, i.e. become a+1. If the embedded data is bit 0, then the difference value is remained. When the difference value of pixel is increased by one, the gray level value of the pixel needs to change according to the relationship of sorted order between the pixel and the base pixel. Record a and b as *OI*.

Let P_{ij} is the *j-th* row and *i-th* column pixel in *CI*, P_b is the selected base pixel, $g(P_{ij})$ denotes the gray level value of P_{ij}, and $d(P_{ij})$ denotes the difference value of P_{ij} and is calculated as $d(P_{ij})=|g(P_{ij})-g(P_b)|,\ P_{ij} \neq P_b$. Let $o(P_{ij})$ is the sorted order of P_{ij} in a block, $c(P_{ij})$ is the change of $g(P_{ij})$ when shifting and embedding, *e* is the embedded bit. As shown in Figure 2, the sorted pixels in a block are divided into several pixel sets. The definitions, process and $c(P_{ij})$ of the pixel sets are listed in Table 1. After *EP*, the changed *CI* is saved as stego image (called *SI*).

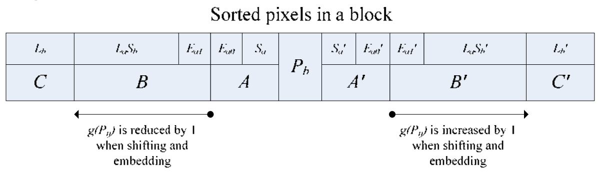

Figure 2. Pixel sets in a block and how to change their gray level value when shifting and embedding

TABLE 1.
DEFINITIONS, PROCESS AND $C(P_{ij})$ OF THE PIXEL SETS IN A BLOCK IN *EP*

Definition	Process	$o(P_{ij})<o(P_b)$			$o(P_{ij})>o(P_b)$		
		Pixel Set		$c(P_{ij})$	Pixel Set		$c(P_{ij})$
$\{ P_{ij} \mid 0 \leqq d(P_{ij}) \leqq a-1 \}$	nothing	A	S_a	0	A′	S_a'	0
$\{ P_{ij} \mid d(P_{ij})=a \text{ and } e=0 \}$	embed 0	A	E_{a0}	0	A′	E_{a0}'	0
$\{ P_{ij} \mid d(P_{ij})=a \text{ and } e=1 \}$	embed 1	B	E_{a1}	-1	B′	E_{a1}'	+1
$\{ P_{ij} \mid a+1 \leqq d(P_{ij}) \leqq b-1 \}$	shift	B	L_aS_b	-1	B′	L_aS_b'	+1
$\{ P_{ij} \mid b \leqq d(P_{ij}) \leqq 255 \}$	nothing	C	L_b	0	C′	L_b'	0

2) Extraction process

In *E'P*, *SI* is divided into non-overlapping $w \times h$ blocks. Then the block difference is calculated using the same base pixels in *EP*. In the difference histogram, the embedded data is extracted from the pixels have difference value *a* (extracted data is bit 0) and *a+1* (extracted data is bit 1), and the area between *a+1* and *b* are shifted back to left by one. When the difference value of pixel is reduced by one, the gray level value of the pixel needs to change according to the relationship of sorted order between the pixel and the base pixel.

Let $c'(P_{ij})$ is the change of $g(P_{ij})$ when extracting and shifting back. As shown in Figure 3, the sorted pixels in a block are divided into several pixel sets. The definitions, process and $c'(P_{ij})$ of the pixel sets are listed in Table 2. After extracting, shifting back, and recovering *LBP* and *UBP*, the changed *SI* is saved as a copy of *CI*.

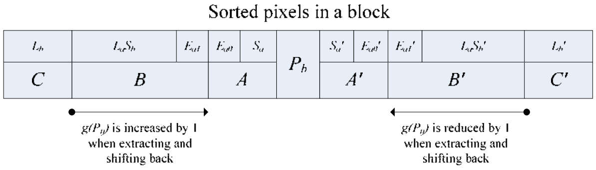

Figure 3. Pixel sets in a block and how to change their gray level value when extracting and shifting back

TABLE 2.
DEFINITIONS, PROCESS AND $C'(P_{IJ})$ OF THE PIXEL SETS IN A BLOCK IN *E'P*

Definition	Process	$o(P_{ij})<o(P_b)$			$o(P_{ij})>o(P_b)$		
		Pixel Set		$c'(P_{ij})$	Pixel Set		$c'(P_{ij})$
$\{ P_{ij} \mid 0 \leqq d(P_{ij}) \leqq a-1 \}$	nothing	A	S_a	0	A'	S_a'	0
$\{ P_{ij} \mid d(P_{ij})=a \}$	extract 0		E_{a0}	0		E_{a0}'	0
$\{ P_{ij} \mid d(P_{ij})=a \}$	extract 1 shift back	B	E_{a1}	+1	B'	E_{a1}'	-1
$\{ P_{ij} \mid a+1 \leqq d(P_{ij}) \leqq b-1 \}$	shift back		L_aS_b	+1		L_aS_b'	-1
$\{ P_{ij} \mid b \leqq d(P_{ij}) \leqq 255 \}$	nothing	C	L_b	0	C'	L_b'	0

B. Interleaving Max-Min Difference

The proposed offset distortion method, embedding process, and extraction process are described below.

1) Offset Distortion Method

The calculated difference when the maximum pixel in every block is designated the base pixel is called the max difference. The calculated difference when the minimum pixel in each block is chosen as the base pixel is called the min difference. The change of the pixel gray level upon shifting and embedding obtained using the max difference is contrary to that obtained using the min difference. The method of interleaving uses the max difference and the min difference to counteract the change in the gray levels of some pixels.

After dividing *CI* into blocks, let *P* is a pixel and *o(P)* denotes the sorted order of *P* in a block. Let $P_m = \arg\max_P(o(P))$, then P_m is called the maximum pixel in the block; let $P_{m'} = \arg\min_P(o(P))$, then $P_{m'}$ is called the minimum pixel in the block. As shown in Figure 4, P_m is selected as base pixel in EP_{2k+1} and $P_{m'}$ is selected as base pixel in EP_{2k+2}. The changes of gray level values of pixels in a block are the same as the definition in Table 1. Let $c''(P_{ij})$ is the change of $g(P_{ij})$ after EP_{2k+1} and EP_{2k+2}, and $c''(P_{ij})$ of pixels in the pixel set intersections are shown in Table 3. The change of $g(P_{ij})$ after EP_{2k+1} and EP_{2k+2} in the intersection of *B* and *B'* is zero.

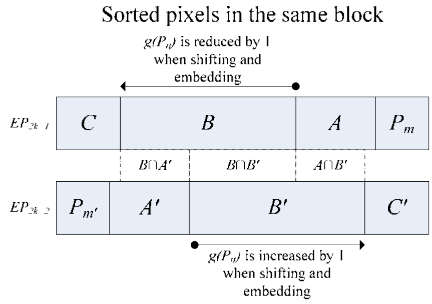

Figure 4. Pixel sets in a block and their intersection in EP_{2k+1} and EP_{2k+2}

TABLE 3.
$C''(P_{IJ})$ OF PIXELS IN THE PIXEL SET INTERSECTIONS IN EP_{2K+1} AND EP_{2K+2}

Pixel Set	$c''(P_{ij})$
$B \cap P_{m'}$ if $C=\{\}$	-1
$B \cap A'$	-1
$B \cap B'$	0
$A \cap B'$	+1
$B' \cap P_m$ if $C'=\{\}$	+1

2) Embedding Process

Figure 5 displays EP in the proposed scheme. Secret Data (called SD) are embedded into CI by repeatedly performing the embedding process until SD have all been fully embedded; finally, SI and key are output. The key contains the information that must be used by the receiver to extract SD. EP has three steps, which are as follows.

Step 1.

Handle *LBP* and *UBP* in *CI*. Use the method described in Section 2.1.1, and save *LUBPI* as *OI*.

Step 2.

Let EPk be the k-th EP. If k is odd, divide CI into blocks and in each block select the maximum pixel (defined in Section 2.2.1) as base pixel to calculate max difference. Find a and b (defined in Section 2.1.1), and save w,h,a,b as OI.

If k is even, use the block size in EPk-1 to divide CI and in each block select the minimum pixel (defined in Section 2.2.1) as base pixel to calculate min difference. Find a and b, and save w,h,a,b as OI.

Step 3.

Use the block difference to embed *SD* and *OI*. Use the method described in Section 2.1.1.

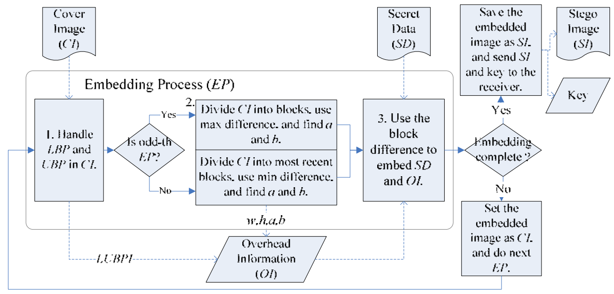

Figure 5. Proposed embedding process

As displayed in Figure 6, the block with pixels P_{44} in Baboon is adopted as an example of the interleaving max-min difference in EP_1 and EP_2. Max difference and min difference are used in EP_1 and EP_2, respectively. Figure 6 displays the results of the sorting of pixels; the calculation of difference values; shifting and embedding in the block in EP_1 and EP_2; and the change in the gray levels in the block upon EP_1 and EP_2. The gray levels of six pixels are changed in Ep_1, eight are changed in EP_2 and only four in EP_1 and EP_2. Therefore, the interleaving max-min difference method effectively reduces the change in the gray levels of the pixels.

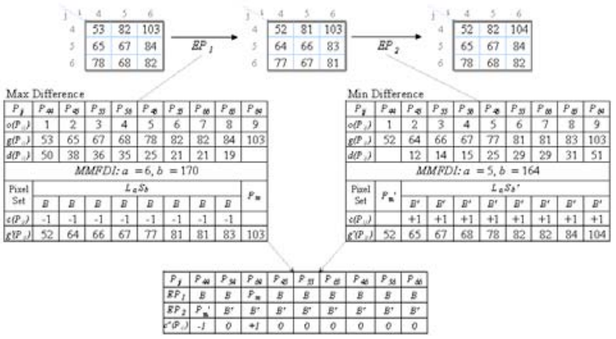

Figure 6. Example for a block of the Baboon image using interleaving max-min difference in EP_1 and EP_2

3) Extraction Process

Figure 7 displays the proposed extraction process. The data are extracted from the *SI* by repeatedly using the extraction process until the data have all been extracted, and a copy of *CI* and *SD* are output. The extraction process has the following three steps.

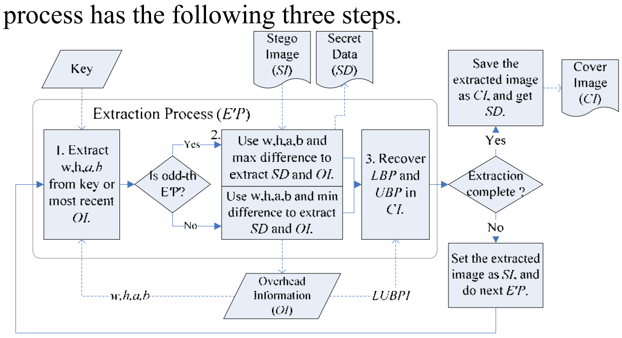

Figure 7. Proposed extraction process

Step 1.

Let $E'P_k$ be the *E'P* contrast with EP_k. Extract *w,h,a,b* from key if *k*=1, and most recent *OI* if *k>1*.

Step 2.

If *k* is odd, use *w,h,a,b* and max difference to extract data. If *k* is even, use *w,h,a,b* and min difference to extract data. The extracting method is described in Section 2.1.2.

Step 3.

Use *LUBPI* to recover *LBP* and *UBP* in *CI*.

As displayed in Figure 8, the block with pixels P_{44} in Baboon is adopted as an example of interleaving max-min difference in $E'P_1$ and $E'P_2$. Min difference and max difference are used in $E'P_1$ and $E'P_2$, respectively. The figure displays the results of the sorting of pixels, the calculation of difference values, shifting back, and extraction in the block in $E'P_1$ and $E'P_2$ and change in the gray levels in the block upon $E'P_1$ and $E'P_2$. The gray levels of the pixels in the block after extraction and shifting back (Figure 8) are the same as that in Baboon *CI*.

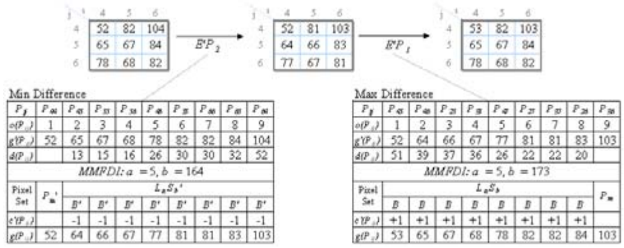

Figure 8. Example for a block of the Baboon image using max-min difference in $E'P_1$ and $E'P_2$

C. Overhead information and key

Figures 9 and 10 display the processing of *OI* and key in *EP* and *E'P*, respectively. *N* is the number of *EP*; EP_k is the *k-th* EP; $E'P_k$ is the *E'P* contrast with EP_k; SD_k is the *SD* embedded in EP_k; OI_k is the *OI* embedded in EP_k; $LUBPI_k$ and $(w,h,a,b)_k$ are *OI* generated in EP_k; marks k.1, k.2 and k.3 indicate the steps in EP_k, and C_N is the number of bits embedded in EP_k. The contents of the overhead information and key are defined below:

$$OI_1=LUBPI_1$$

$$OI_k=LUBPI_k+(w,h,a,b)_{k-1},\ 2\leq k\leq N$$

$$key=(w,h,a,b)_N+N+C_N$$

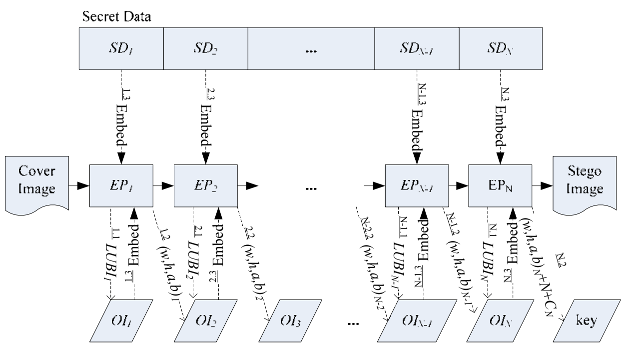

Figure 9. Processing of OI and key in EP

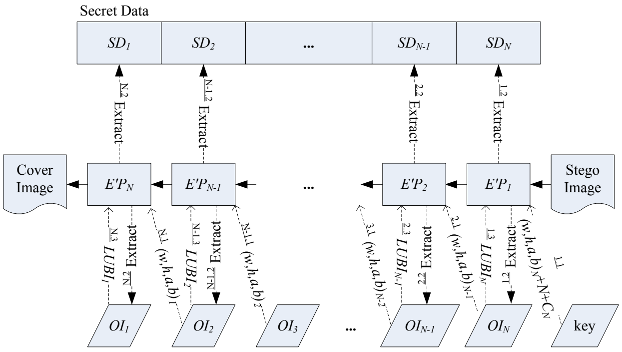

Figure 10. Processing of OI and key in E′P

III. EXPERIMENTAL RESULTS

The methods of Ni et al. [15], F&S4 (Fallahpour and Sedaaghi with 4 blocks) [16], F&S16 (Fallahpour and Sedaaghi with 16 blocks) [16], L&H (Lin and Hsueh) [17], fixed site difference and the proposed scheme were implemented in the Java programming language. The performance of each method was tested using ten 512×512 grayscale images as cover images and part of a 512×512 grayscale image as secret data, as shown in Figure 11. Performance was evaluated using the payload (size of embedded secret data) and PSNR (peak signal to noise ratio). A higher payload and PSNR represents better performance.

Figure 11. 512×512 grayscale cover images and secret data image tested in experiments

The performance of the proposed method was evaluated using various w and h ($1 \leqq w,h \leqq 4$, $w \times h \geqq 3$). This embedding process was repeated until the PSNR approached 30dB. Table 4 presents the results of this process, where the unit of payload is bpp (bits per pixel). The 3×4 block yielded the largest average payload, 1.120bpp.

The performance was measured and compared with those of Ni [15], F&S4 [16], F&S16 [16] and L&H [17]. The embedding process was repeated until the PSNR approached 30dB. Table 5 presents the results of this process, where the unit of payload is bpp. For each image and on average, the proposed method outperformed the other methods.

The comparison of performance (the payloads when the PSNR approached 30dB) of proposed methods with that of histogram shifting category methods under different cover images is shown as Figure 12. The figure shows that our proposed methods perform better than other methods.

TABLE 4.
PERFORMANCE OF PROPOSED METHOD USING VARIOUS BLOCK SIZES

block size	3		4			6		8		9	12		16
CI \ w×h	1x3	3x1	1x4	2x2	4x1	2x3	3x2	2x4	4x2	3x3	3x4	4x3	4x4
Airplane	1.143	1.167	1.248	1.346	1.289	1.426	1.443	1.460	1.494	1.476	1.513	1.520	1.494
Baboon	0.436	0.517	0.482	0.558	0.541	0.569	0.577	0.585	0.594	0.602	0.585	0.585	0.576
Barbara	0.744	0.644	0.796	0.814	0.715	0.878	0.837	0.899	0.852	0.882	0.914	0.899	0.878
Boat	1.038	0.957	1.114	1.107	1.046	1.182	1.192	1.222	1.256	1.258	1.266	1.279	1.288
GoldHill	0.736	0.781	0.767	0.840	0.833	0.875	0.884	0.860	0.887	0.895	0.883	0.904	0.907
Lena	0.991	0.870	1.077	1.044	0.917	1.148	1.097	1.175	1.105	1.120	1.183	1.125	1.166
Pepper	0.970	1.000	1.018	1.138	1.019	1.187	1.179	1.183	1.177	1.178	1.170	1.123	1.167
Sailboat	0.749	0.758	0.802	0.859	0.809	0.923	0.914	0.943	0.971	0.961	0.970	0.968	0.960
Tiffany	1.141	1.038	1.209	1.241	1.107	1.307	1.282	1.376	1.346	1.365	1.383	1.339	1.367
Toy	1.089	1.060	1.156	1.179	1.156	1.261	1.245	1.263	1.293	1.290	1.333	1.332	1.359
average	0.904	0.879	0.967	1.013	0.943	1.076	1.065	1.097	1.098	1.103	1.120	1.107	1.116

The maximum payload of each image

TABLE 5.
COMPARISON OF PERFORMANCE OF NI, F&S4, F&S16, L&H, AND PROPOSED METHOD

Images	Ni [15]	F&S4 [16]	F&S16 [16]	L&H [17]	Proposed(3x4)
Airplane	0.48	0.62	0.77	0.81	1.51
Baboon	0.18	0.20	0.23	0.38	0.58
Barbara	0.16	0.17	0.25	0.44	0.91
Boat	0.34	0.39	0.54	0.64	1.27
GoldHill	0.15	0.21	0.28	0.56	0.88
Lena	0.17	0.26	0.38	0.62	1.18
Pepper	0.18	0.21	0.34	0.73	1.17
Sailboat	0.30	0.31	0.48	0.53	0.97
Tiffany	0.36	0.49	0.60	0.76	1.38
Toy	0.59	0.64	0.84	0.70	1.33
average	0.29	0.35	0.47	0.62	1.12

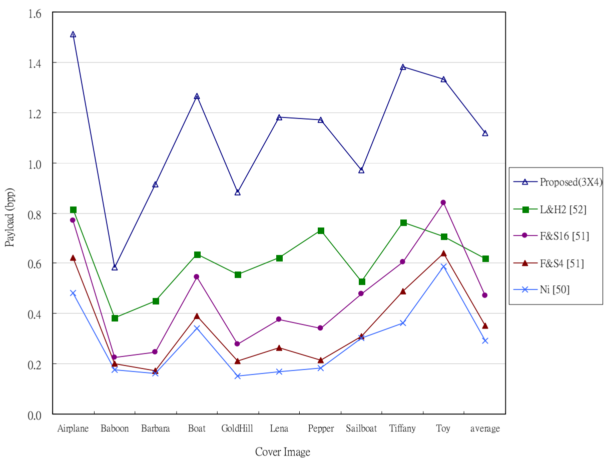

Figure 12. Comparison of performance of proposed methods with that of histogram shifting category methods under different cover images

Figures 13, 14, 15 and 16 compare the average performance (payload and PSNR value) of the proposed methods with those of the histogram shifting, data compression, difference expansion and integer wavelet transform category methods, respectively. The figures reveal that the proposed method outperforms other methods.

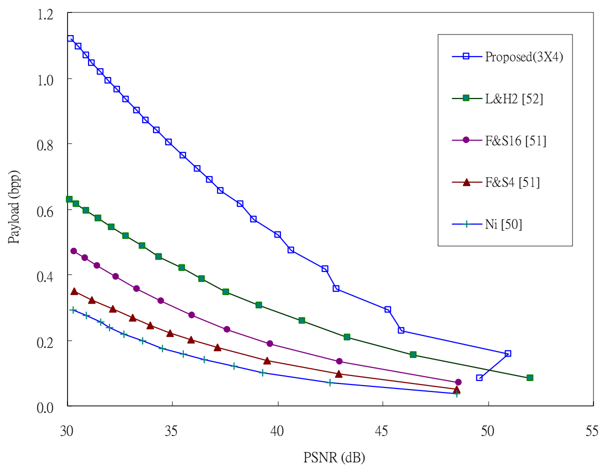

Figure 13. Comparison of average performance of proposed method with those of histogram shifting category methods

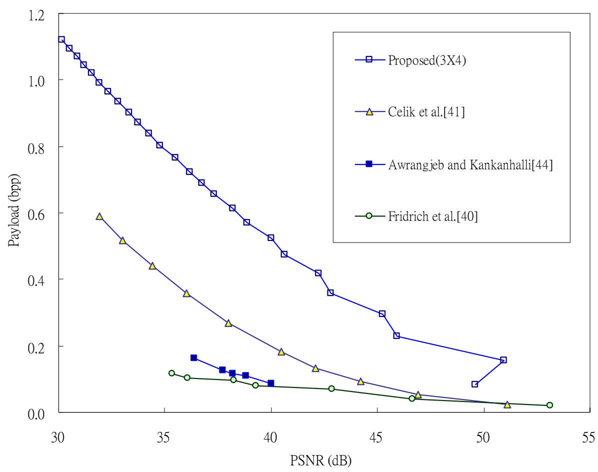

Figure 14. Comparison of average performance of proposed method with those of data compression category methods

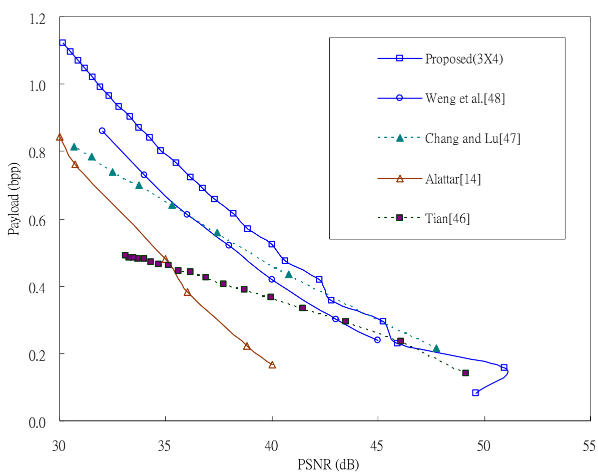

Figure 15. Comparison of average performance of proposed method with those of difference expansion category methods

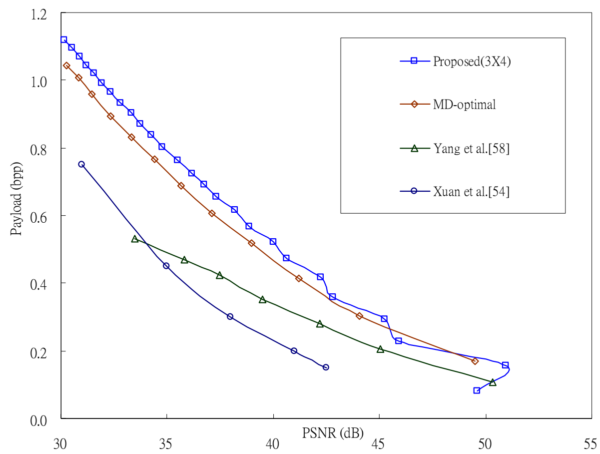

Figure 16. Comparison of average performance of proposed method with those of integer wavelet transform category methods

IV. CONCLUSIONS

This paper proposed a block difference histogram method and an offset distortion method to embed data with reversible. In the experiment, when the PSNR approached 30dB, the optimal payloads of the interleaving max-min difference histogram method were 1.120bpp. Experimental results reveal that the proposed method outperforms many other reversible data hiding methods.

REFERENCES

[1] M. Awrangjeb, “An overview of reversible data hiding,” *International Conference on Computer and Information Technology (ICCIT)*, 2003, pp. 75-79.

[2] Y.Q. Shi, Z. Ni, D. Zou, C. Liang, and G. Xuan, “Lossless data hiding: fundamentals, algorithms and applications,” *IEEE International Symposium on Circuits and Systems* (*ISCAS*), Vol. II, 2004, pp. 33-36.

[3] J. Fridrich, M. Goljan, and R. Du, “Invertible authentication,” *Proc. SPIE Security and Watermarking of Multimedia Contents*, 2001, pp. 197–208.

[4] J. Fridrich, M. Goljan, R. Du, “Lossless data embedding-new paradigm in digital watermarking,” *EURASIP Journal on Applied Signal Processing*, Vol. 2, 2002, pp. 185–196.

[5] M.U. Celik, G. Sharma, A.M. Tekalp, and E. Saber, ”Reversible data hiding,” *Proc. IEEE Int. Conf. Image Processing*, Vol. 2, 2002, pp. 157-160.

[6] M.U. Celik, G. Sharma, A.M. Tekalp, and E. Saber, “Lossless generalized-LSB data embedding,” *IEEE Trans. Image Processing*, Vol. 14, No.2, 2005, pp. 253–266.

[7] M.U. Celik, G. Sharma, A.M. Tekalp, “Lossless watermarking for image authentication: a new framework and an Implementation,” *IEEE Trans. Image Processing*, Vol. 15, No.4, 2006, pp. 1042–1049.

[8] M. Awrangjeb and M.S. Kankanhalli, “Lossless Watermarking Considering the Human Visual System,” *Lecture Notes in Computer Science*, Vol. 2939, 2004, pp. 329-336.
[9] M. Awrangjeb and M.S. Kankanhalli, “Reversible watermarking using a perceptual model,” *Journal of Electronic Imaging*, Vol. 14, No. 1, 013014, 2005, pp. 1-8.
[10] J. Tian, “Reversible data embedding using a difference expansion,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No.8, 2003, pp. 890–896.
[11] A.M. Alattar, “Reversible watermark using the difference expansion of a generalized integer transform,” *IEEE Trans. Image Processing*, Vol. 13, No.8, 2004, pp. 1147–1156.
[12] C.C. Chang, and T.C. Lu, “A difference expansion oriented data hiding scheme for restoring the original host images,” *The Journal of Systems and Software*, Vol. 79, 2006, pp. 1754–1766.
[13] S. Weng, Y. Zhao, J.S. Pan, R. Ni, “A novel reversible watermarking based on an integer transform,” *IEEE Int. Conf. Image Processing*, Vol. 3, 2007, pp. 241-244.
[14] S. Weng, Y. Zhao, J.S. Pan, and R. Ni, “Reversible watermarking based on invariability and adjustment on pixel pairs,” *IEEE Signal Processing Letters*, Vol. 15, 2008, pp. 721-724.
[15] Z. Ni, Y.Q. Shi, N. Ansari, and W. Su, “Reversible data hiding,” *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 16, No. 3, 2006, pp. 354-362.
[16] M. Fallahpour and M.H. Sedaaghi, “High capacity lossless data hiding based on histogram modification,” *IEICE Electronics Express*, Vol. 4, No. 7, 2007, pp. 205-210.
[17] C.C. Lin and N.L. Hsueh, “Hiding data reversibly in an image via increasing differences between two neighboring pixels,” *IEICE TRANS. INF. & SYST.*, Vol. E90–D, No. 12, 2007, pp. 2053-2059.
[18] G. Xuan, J. Chen, J. Zhu, Y.Q. Shi, Z. Ni, and W. Su, “Lossless data hiding based on integer wavelet transform,” *IEEE Workshop on Multimedia Signal Processing*, 2002, pp. 9-11.
[19] G. Xuan, Y.Q. Shi, Z.C. Ni, J. Chen, C. Yang, Y. Zhen, J. Zheng; “High capacity lossless data hiding based on integer wavelet transform,” in *Pro. Int. Sym. Circuits and Systems*, Vol. 2, 2004, pp. 29-32.
[20] G. Xuan, Y.Q. Shi, C. Yang, Y. Zheng, D. Zou, snd P. Chai, “Lossless Data Hiding Using Integer Wavelet Transform and Threshold Embedding Technique,” *IEEE Int. Conf. Multimedia and Expo*, 2005, pp. 1520-1523.
[21] L. Yang, P. Hao, and C. Zhang, “Progressive reversible data hiding by symmetrical histogram expansion with piecewise-linear haar transform,” *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. II, 2007, pp. 265-268.

Hsien-Wei Yang is an Assistant Professor of Department of Information Management at Overseas Chinese University, Taiwan, ROC. His research interests include Data Hiding, Web Service, and E-Learning. From 1999 to date, he was a faculty of the Information Management Department at Oversea Chinese University. He received his Ph.D. in Applied Mathematics from ChungHsing University in 2009. The contact address of Dr. Yang is Department of Information Management, Overseas Chinese University, No:100, Chiao Guang Rd., Taichung 407, Taiwan, R.O.C. E-mail: fiber@ocu.edu.tw.

Kuo-Feng Hwang is an Assistant Professor of Department of Information Management at Overseas Chinese University, Taiwan, ROC. His research interests include Internet Security, Data Mining, Distance Learning, and E-Commerce. From 1999 to date, he was a faculty of the Information Management Department at Oversea Chinese University. He received his Ph.D. in Applied Mathematics from ChungHsing University in 2006. The contact address of Dr. Hwang is Department of Information Management, Overseas Chinese University, No:100, Chiao Guang Rd., Taichung 407, Taiwan, R.O.C. E-mail: kfhwang@ocu.edu.tw.

Shou-Shiung Chou is a Lecturer of Department of Hospitality Management at Tunghai University, Taiwan, ROC. His research interests include Marketing Services and Hotel Management. He is studying doctor degree in Department of Business Administration, National Taipei University. After Mr. Chou finished the Certificate in Hotel Management in Singapore International Hotel And Tourism College, he went as a Hotelier in Evergreen Laurel Hotel Hong Kong and Taichung for seven years..