High-Performance Reversible Data Hiding with Overflow/Underflow Avoidance

Ching-Yu Yang and Wu-Chih Hu

This paper proposes reversible data hiding using minimum/maximum preserved overflow/underflow avoidance (MMPOUA). The proposed MMPOUA algorithm consists of three main steps. These steps include the minimum (or maximum) pixel fixing, pixel squeezing, and pixel isolation. The aims of pixel fixing are to keep the minimum (or maximum) pixel of a host block unchanged and prevent the occurrence of overflow/underflow. Both the pixel squeezing and pixel isolation supply hiding storage while keeping the amount of distortion low. The proposed method can avoid (or significantly reduce) the overhead bits used to overcome overflow/underflow issues. At an embedding rate of 0.15 bpp, the proposed algorithm can achieve a PSNR value of 48.52 dB, which outperforms several existing reversible data hiding schemes. Furthermore, the algorithm performed well in a variety of images, including those in which other algorithms had difficulty obtaining good hiding storage with high perceived quality.

Keywords: Reversible data hiding, MMPOUA algorithm.

Manuscript received Sept. 8, 2010; revised Dec. 3, 2010; accepted Jan. 11, 2011.

Ching-Yu Yang (phone: +886 69264115, email: chingyu@npu.edu.tw) and Wu-Chih Hu (email: wchu@npu.edu.tw) are with the Department of Computer Science and Information Engineering, National Penghu University of Science and Technology, Penghu, Taiwan.

doi:10.4218/etrij.11.0110.0534

I. Introduction

With the proliferation of high-speed networks and high-capacity digital recording devices, individuals (and organizations) can now conveniently and economically exchange information over the Internet. Keeping data free from being tampered with or eavesdropped on and preventing malicious intervention in electronic communication have become important issues. In addition to encryption/decryption systems, data hiding can assist in solving these problems. Key data hiding applications include proof of ownership, authentication, and covert message transfer. Data hiding can be generally divided into two categories: steganography and watermarking [1], [2]. In steganography [3], [4], the hidden message is often unrelated to the host media; however, the embedding capacity and resulting perceived quality are the primary aims of the authors. In watermarking [5], [6], the embedded message is related to the media and conveys additional information about the media. Robustness is the key feature of watermarking schemes.

The data hiding techniques mentioned above are irreversible. As host media, such as medical and military images, geographic systems, and satellite resources are often very valuable, sustaining the host media following data extraction is imperative. Several researchers [7]-[23] have recently introduced lossless data hiding for this purpose. Five classifications of these methods are examined in the following subsections.

1. Difference Expansion Techniques

Tian [7] used a difference expansion (DE) technique to derive high capacity, low-distortion reversible data hiding. This technique divided the image into pairs of pixels, and a secret message was embedded into the difference between the pixels of each pair that was not expected to cause overflow/underflow. Simulation showed that the payload size and the perceived quality of the marked images generated by the technique were optimal results among existing research at that time. In single pass embedding, the optimal payload of this technique was 0.5 bits per pixel (bpp). Alattar [8] used DE with vectors instead of pixel pairs to extend and improve the performance of Tian's algorithm. In a single pass, Alattar's algorithm could embed several bits into every vector. Applications of this algorithm could recursively occur across both grayscale and color images. Weng and others [9] suggested an algorithm for lossless data hiding based on the invariability of the sum of pixel pairs and pair-wise difference adjustment (PDA). Higher peak-signal-to-noise ratio (PSNR) can be achieved by making smaller modifications to pixel pairs. In addition, PDA was proposed to significantly reduce the overhead size. To increase hiding capacity, Hu and others [10] proposed a variant DE-based technique that improved the compressibility of the location map. Compared to conventional DE-based schemes, their technique provided increased embedding storage and performed well with a variety of image types.

2. Histogram-Based Schemes

Ni and others [11] utilized the zero or minimum points of the histogram of an image and slightly modified the pixel values to embed data bits into the images without any loss. The PSNR generated by their method exceeded 48 dB. Lin and others [12] proposed a multilayer scheme for reversible data hiding based on modification of the difference histogram. By combining the peak point of a difference image concept with a multilevel hiding strategy, the scheme maintained high capacity while keeping distortion low. In first-level embedding, the low bound of PSNR reached 42.69 dB. Tai and others [13] used the histogram modification technique with a binary tree structure. In addition, a histogram shifting technique was explored to prevent overflow/underflow. By using the difference between adjacent pixels rather than a single pixel, their algorithm operated at high capacity while maintaining low distortion. Based on modification of the difference histogram between subsampled images, Kim and others [14] developed an efficient lossless data hiding algorithm. The algorithm shifted the difference histogram and then embedded data bits into the modified pixel values. The algorithm prevented any overflow/underflow issues and did not require overhead information during data extraction. Such algorithms normally reach a payload size of 1.0 bpp. By using the interleaving maximum/minimum difference histogram with the shifting technique, Yang and others [15] achieved a high-capacity reversible data hiding method. Owing to the interchangeable use of max-difference and min-difference, the distortion can be significantly reduced. Simulation indicated that the optimal embedding rate provided by the method was 1.120 bpp, with a PSNR value of approximately 30 dB.

3. Prediction-Based Methods

Based on prediction-error expansion and the histogram shifting technique, Thodi and Rodriguez [16] presented an effective and reversible method of data hiding. Simulation showed that prediction-error expansion doubled the maximum embedding capacity as compared to difference expansion. The perceived quality of the marked images was good at a moderate embedding capacity. Tsai and others [17] presented a lossless data hiding approach using predicted coding and histogram shifting. The prediction technique first explored the similarities of neighboring pixels in the image, after which the residual histogram of the predicted errors of the host image hid the data bits. Compared to a conventional histogram-based method, the resulting PSNR improved by approximately 1.5 dB when embedding the same amount of data bits. Yang and others [18] suggested a reversible data hiding method by using interleaving prediction. All predictive values were transformed into histogram to generate high peak values and improve hiding capacity. For single-layer embedding, the average PSNR of the marked images was larger than 48 dB. Lee and others [19] developed an adaptive reversible data hiding approach based on the prediction of difference expansion. Simulation demonstrated that a high perceived quality of the marked image can be achieved by the approach. Moreover, the location map was not required during data extraction.

4. Robustness-Oriented Approaches

Ni and others [20] developed a robust lossless data hiding technique based on the patchwork theory, the distribution features of pixel groups, error codes, and the permutation scheme. Although the payload size of the technique did not exceed 1,024 bits, the marked images generated by the technique contained no salt-and-pepper noise and the resulting PSNR exceeded 38 dB. Additionally, the marked images were robust to JPEG/JPEG2000 compression. By shifting the mathematical difference values of a block, Zeng and others [21] designed a lossless and robust data hiding method. The data bits were embedded into blocks by shifting mathematical difference values. Due to the separation of the bit-0-zone and the bit-1-zone, as well as the particularity of mathematical difference, the method was tolerant of non-malicious JPEG compression to some extent. The resulting images, as compared to the images produced by the technique of Ni and others [20], showed that the method of Zeng and others [21] increased embedding capacity but at the sacrifice of bit error rate and perceived quality.

5. Human Visual System

By considering the human visual system (HVS), Awrangjeb and Kankanhalli [22], [23] presented a novel reversible data hiding algorithm. Because the message was embedded into the host image with consideration of the HVS, the resulting marked images contained no perceptible artifacts. Experiments confirmed that the algorithm provided a higher payload size than other techniques at the time.

The majority of the above methods share the potential problem of overflow/underflow. Namely, these methods require a large amount of overhead bits to deal with overflow/underflow during bit embedding. This paper proposes a reversible data hiding technique based on the minimum/maximum preserved overflow/underflow avoidance (MMPOUA) algorithm, which can effectively overcome the overflow/underflow issue. In addition, the proposed method provides high capacity while maintaining low distortion.

The rest of the paper is organized as follows. Section II describes the proposed method, including the minimum preserved algorithm, the maximum preserved algorithm, prevention of overflow/underflow, and analysis of overhead. Section III presents the simulation results and the comparisons of performance. Section IV provides the conclusion.

II. Proposed Method

The proposed MMPOUA method consists primarily of the minimum preserved and maximum preserved algorithms. Normally, the minimum preserved algorithm embeds a secret message into a host image. However, the maximum preserved algorithm replaces the minimum preserved algorithm under either of the following conditions: (i) the minimum preserved algorithm is incapable of conducting reversible data hiding on a certain image or (ii) the payload size generated by the minimum preserved algorithm using the integer k with a value of 2 is less than that generated using the integer k with a value of 1. The following sections provide the specifics of the proposed method.

1. Minimum Preserved Algorithm

As described in the abstract, the minimum preserved algorithm consists of three major steps: minimum pixel fixing, pixel squeezing, and pixel isolation. The first step determines the minimum value of a host block and subtracts the rest of the pixels from the minimum one. The second step shifts the pixels to a new value when their value is larger than γ. Finally, to maintain low distortion, a reduced pixel (and a shifted pixel) becomes "isolated" when its value is greater than or equal to β. This means that the data bits cannot hide within the isolated pixels. Modulo-2^k substitution is then used to embed data bits into the "qualified" pixels. Both β and γ are control parameters. The following paragraphs summarize the idea behind the encoder for the minimum preserved algorithm.

Let $C=\{p_{ij}\}_{i=0}^{(n\times n)-1}$ be the j-th non-overlapping block with a size of $n\times n$ that the algorithm divides from a host image. Also, let $p_{\min}=\arg\mathrm{Min}\{p_{ij}\}_{i=0}^{(n\times n)-1}$ be the minimum pixel in a host block. This study used $\{\hat{p}_{ij}\}_{i=0}^{(n\times n)-1}=\{p_{ij}\}_{i=0}^{(n\times n)-1}-p_{\min}$ to measure the reduced pixels in the block. The algorithm then performs both pixel squeezing and pixel isolation using $\{\hat{p}_{ij}\}_{i=0}^{n\times n-1}$. More specifically, $\hat{p}_{ij}$ shifts to a new value, $\tilde{p}_{ij}=\hat{p}_{ij}-\gamma$, if it satisfies $\hat{p}_{ij}>\gamma$. It is necessary to use a bitmap here to flag whether or not a pixel of the block undergoes adjustment. To maintain low levels of distortion, the isolation process subsequently conducts to $\hat{p}_{ij}$ and $\tilde{p}_{ij}$, in accordance with the following rule:

$$\breve{p}_{ij}=\widehat{p}_{ij}+(2^k-1)\beta,\ \text{ if }\ \widehat{p}_{ij}\geq\beta, \tag{1}$$

where $\widehat{p}_{ij}\in\{\hat{p}_{ij},\tilde{p}_{ij}\}$. This means that the isolated pixels $\breve{p}_{ij}$ do not carry the data bits. After adjustments to the data bits b_s, they are ready for embedding into the adjusted blocks. The reduced (or shifted) pixels c_r, which satisfy $0\leq c_r<\beta$, are multiplied by 2^k to obtain $\hat{c}_r$. The data bits b_s are then added to $\hat{c}_r$. To form a hidden block, $p_{\min}$ is added to $\hat{c}_r$ and $\breve{p}_{ij}$, respectively. This procedure is repeated until all data bits have been processed.

2. Maximum Preserved Algorithm

The process of data embedding for the maximum preserved algorithm is similar to that of the minimum preserved algorithm. The following paragraphs summarize the major steps of the encoder part of the maximum preserved algorithm.

Let $p_{\max}=\arg\mathrm{Max}\{p_{ij}\}_{i=0}^{(n\times n)-1}$ be the maximum pixel of a host block. The reduced pixels of the j-th block are computed by $\{\hat{p}_{ij}\}_{i=0}^{(n\times n)-1}=\{p_{ij}\}_{i=0}^{(n\times n)-1}-p_{\max}$. Then, $\hat{p}_{ij}$ is shifted to a new value, $\tilde{p}_{ij}=\hat{p}_{ij}+\gamma$, if it satisfies $\hat{p}_{ij}<-\gamma$. The isolation process subsequently processes $\hat{p}_{ij}$ and $\tilde{p}_{ij}$ according to the following rule:

$$\breve{p}_{ij}=\widehat{p}_{ij}-(2^k-1)\beta,\ \text{ if }\ \widehat{p}_{ij}\leq-\beta, \tag{2}$$

with $\widehat{p}_{ij}\in\{\hat{p}_{ij},\tilde{p}_{ij}\}$. The reduced (or shifted) pixels c_l, which satisfy $-\beta<c_l<0$, are then multiplied by 2^k to obtain

$\hat{c}_l$. The b_s is then subtracted from $\hat{c}_l$. To form a hidden block, $p_{\max}$ is added to $\hat{c}_l$ and $\breve{p}_{ij}$, respectively. The procedure is repeated until all data bits have been processed.

Figures 1 and 2 depict two examples of bit embedding using the minimum preserved approach and the maximum preserved approach, respectively. The k used here is 1. Both control parameters β and γ are set to 5 and 3, respectively. Figure 1(a) illustrates a host block. Figure 1(b) shows a reduced block introduced by subtracting each pixel in Fig. 1(a) from the minimum 52 of the block. A rectangle marks the squeezed coefficients in Fig. 1(c). Figure 1(d) shows that a pixel becomes "isolated" if it is greater than 5. Note that a gray highlighted number denotes an isolated pixel. Figure 1(e) shows a hidden block. Finally, Fig. 1(f) shows the formation of the marked block achieved by adding the minimum to all of the pixels, except the minimum pixel in Fig. 1(e). The mean square error (MSE) computed from Figs. 1(a) and 1(f) was 5.11. Figure 2 shows another example of bit embedding using the maximum preserved algorithm. Based on the procedures of the algorithm, the reduced block, squeezed block, isolated block, and hidden block are introduced in Figs. 2(b) to 2(e). The resulting MSE for the marked block in Fig. 2(f) was 2.00.

52	54	54
57	59	56
61	55	55

(a)

52	2	2
5	7	4
9	3	3

(b)

52	2	2
[2]	[4]	[1]
[6]	3	3

(c)

52	2	2
2	4	1
11	3	3

(d)

52	4	4
4	9	2
11	7	6

(e)

52	56	56
56	61	54
63	59	58

(f)

Fig. 1. Example of bit embedding using minimum preserved algorithm with input bit-stream of 0001010: (a) 3×3 block, (b) reduced block, (c) squeezed block, (d) isolated block, (e) hidden block, and (f) marked block.

135	128	125
130	123	128
128	130	129

(a)

135	–7	–10
–5	–12	–7
–7	–5	–6

(b)

135	[–4]	[–7]
[–2]	[–9]	[–4]
[–4]	[–2]	[–3]

(c)

135	–4	–12
–2	–14	–4
–4	–2	–3

(d)

135	–8	–12
–4	–14	–9
–9	–5	–6

(e)

135	127	123
131	121	126
126	130	129

(f)

Fig. 2. Example of bit embedding using maximum preserved algorithm with an input bit-stream of 001110: (a) host block, (b) reduced block, (c) squeezed block, (d) isolated block, (e) hidden block, and (f) marked block.

A similar reverse process can present the steps of the decoding parts of the minimum preserved and maximum preserved algorithms. Figure 3 summarizes only the encoding and decoding parts of the minimum preserved algorithm.

3. Prevention of Overflow/Underflow and Overhead Analysis

Normally, when an integer k is set at 1, that is, when modulo-2 substitution is employed in the minimum (or maximum) preserved algorithm, the proposed method can achieve reversible data hiding without any occurrence of overflow/underflow. However, if there is a pixel in a block with a value equal to 255 or 0, overflow/underflow can occur during bit embedding. To overcome this issue, two thresholds, ϕ_1 and ϕ_2, were used in the minimum and maximum preserved algorithms. This means that if a pixel in a block has a value greater than ϕ_1, then the minimum preserved algorithm will skip the block. A skipped block does not hide data bits. Extra overhead is necessary to record the index of the skipped block in a host image. Conversely, if a pixel in a block has a value less than ϕ_2, then the maximum preserved algorithm will skip the block.

In the proposed method, the overhead information used in the process of pixel squeezing without using the block-skipped policy was $\left\lfloor \frac{M}{n} \right\rfloor \times \left\lfloor \frac{N}{n} \right\rfloor \times n^2 \le MN$ bits. However, if the block skipped policy is applied, the extra overhead is $B_S \times N_b$ bits, where B_S and N_b denote the number of skipped blocks and number of bits, respectively, used to record the index of a skipped block in a host image. It is obvious that $\left\lfloor \frac{M}{n} \right\rfloor \times \left\lfloor \frac{N}{n} \right\rfloor < 2^{15}$ if $n \ge 3$. Namely, B_S with a 15-bit length is sufficient to record the index of a skipped block with a size of $n \times n$. As a result, the total overhead of the bit was $\left(\left\lfloor \frac{M}{n} \right\rfloor \times \left\lfloor \frac{N}{n} \right\rfloor \times n^2 \right) + \left(B_N \times B_S \right)$ in the proposed method using the block-skipped policy. However, when $B_S > \frac{1}{N_b} \left\lfloor \frac{M}{n} \right\rfloor \left\lfloor \frac{N}{n} \right\rfloor$, using the block-skipped policy was not feasible in the proposed method. In this case, instead of using B_S and N_b, this study used $\left\lfloor \frac{M}{n} \right\rfloor \left\lfloor \frac{N}{n} \right\rfloor$ as a location map to record the index of the skipped blocks. This resulted in a total overhead of

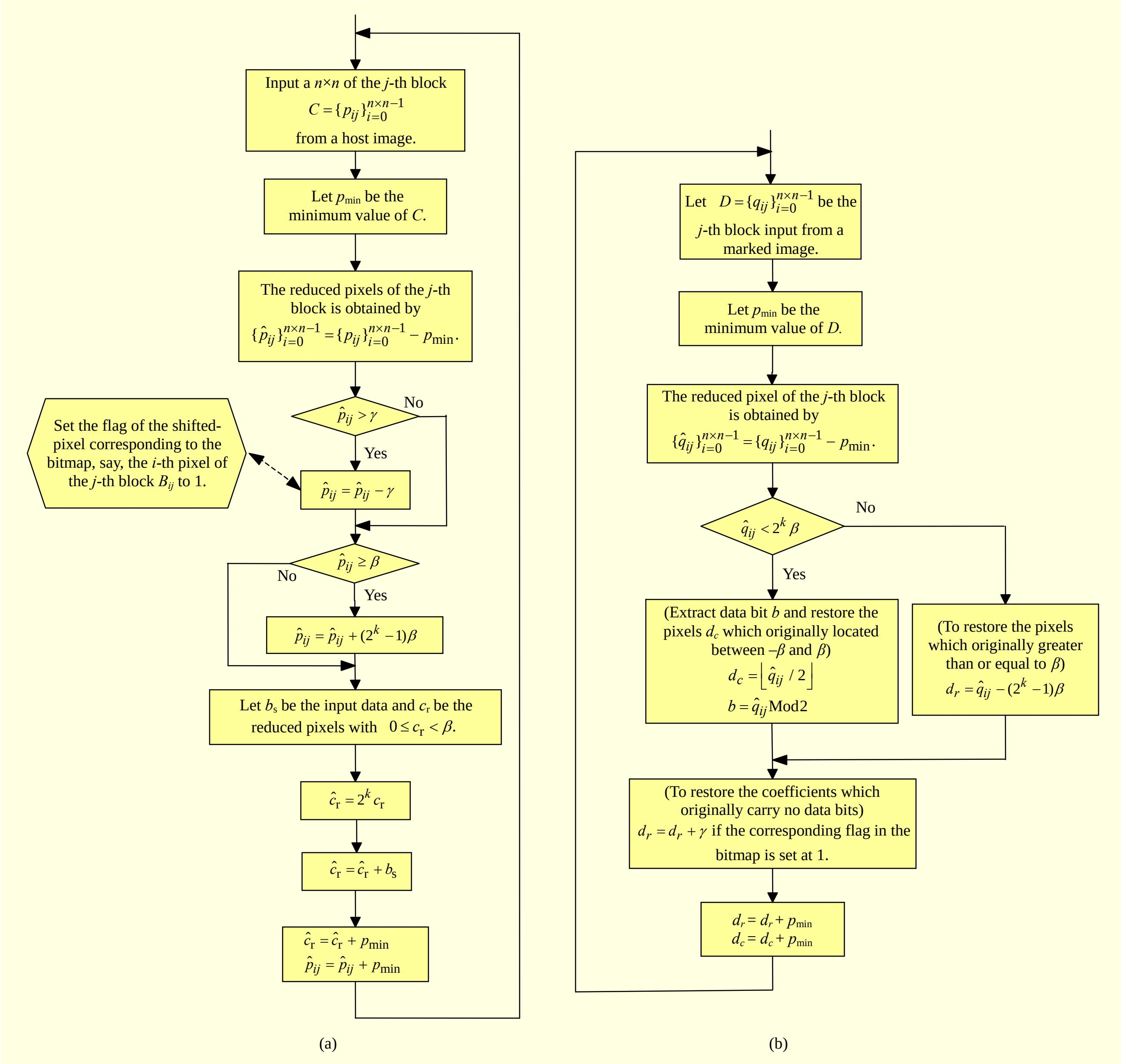


Fig. 3. Flowchart of proposed method: (a) encoder and (b) decoder.

$$\left(\left\lfloor\frac{M}{n}\right\rfloor\times\left\lfloor\frac{N}{n}\right\rfloor\times n^2\right)+\left(\left\lfloor\frac{M}{n}\right\rfloor\times\left\lfloor\frac{N}{n}\right\rfloor\right)=\left(1+\frac{1}{n^2}\right)\left\lfloor\frac{M}{n}\right\rfloor\left\lfloor\frac{N}{n}\right\rfloor$$

bits. To further increase data security and help the receiver to extract the hidden message, the overhead information can be losslessly compressed by using either the run-length coding algorithm or JBIG2 [24]. The resulting bit stream can then sent by an out-of-band transmission to the receiver.

III. Experimental Results

Figure 4 displays several 512×512 grayscale images used in this experiment as the host images. This study used a quarter of the host image Lena as test data. The size of the block was 3×3. To achieve a higher PSNR value, an integer k was set to 1. However, to achieve a higher hiding rate, for example, above 0.80 bpp, k can be set to 2. Figure 5 illustrates the relationship between the PSNR and the payload size for these images. This figure also shows that the hiding rate for all images, except Baboon, exceeded 0.80 bpp. More importantly, the hiding rate for images Lena, Jet, Peppers, and Tiffany could reach 1.32 bpp with a PSNR value of 28.30 dB. Note that the relation between both parameters is $|\beta|\geq|\gamma|>0$. Figure 6 indicates that the payload size varied according to different combinations

Fig. 4. Host images.

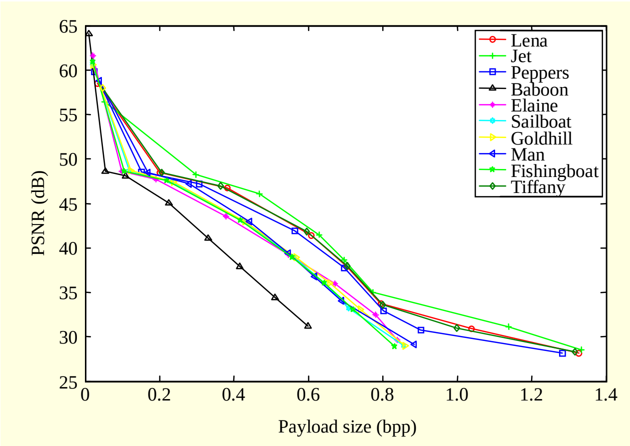


Fig. 5. Tradeoff between PSNR and payload using proposed method in test images.

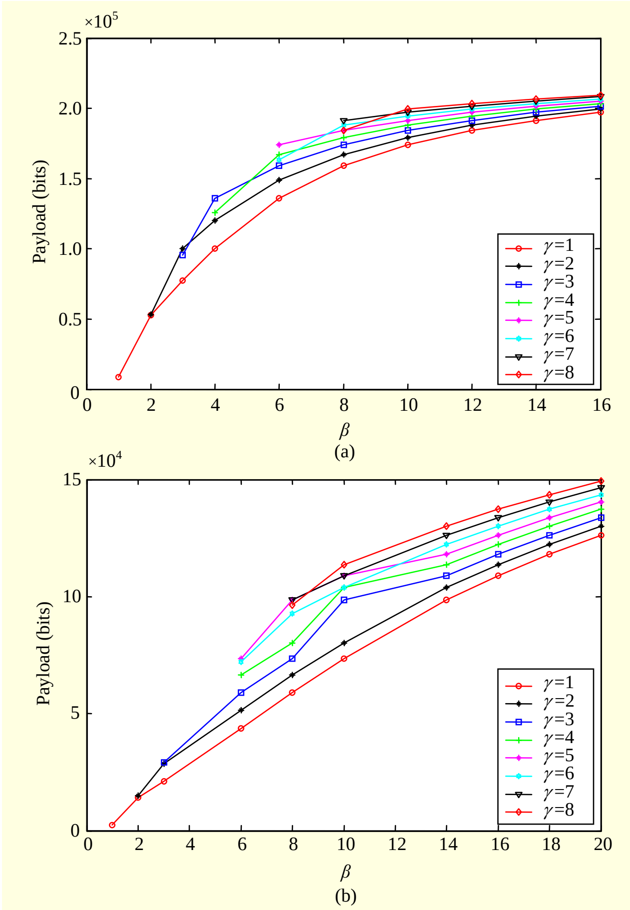


Fig. 6. Relationship between two control parameters β and γ using proposed medthod in the test images: (a) Lena and (b) Baboon.

of these two parameters in the images Lena and Baboon. The experiment shows that when γ is fixed, the payload size gradually grows as β increases. Note that the proposed method can achieve enhanced hiding performance if the combination of β and γ is $\beta = \gamma - 1$. Table 1 lists the PSNR and payload size resulting from using the proposed method with a variety of combinations of β and γ. The average of the maximum PSNR for each image was 59.94 dB with a payload of 7,331 bits; whereas, an average payload size is 191,787 bits with a PSNR value of 33.55 dB. Moreover, the proposed method obtained a higher hiding rate by using k=2, as Table 2 shows. To avoid the occurrence of overflow/underflow, this method skipped a few of host blocks in images, such as Sailboat, Man, and Fishingboat. The N/A in Table 2 implies that the performance of a combination of parameters β and γ, with integer k=2, did not exceeded the performance of these parameters with k=1 in the test image. Figure 7 depicts the marked images generated by the proposed method. It can be seen that the perceived quality of these images were acceptable. Their average PSNR and payload was 29.50 dB and 0.85 bpp, respectively.

This study compared two graceful algorithms, the scheme of Tai and others [13] and the approach of Kim and others [14], with our method. Figure 8 shows performance comparisons between these methods in several images. Figures 8(a) and (b) show that the proposed method displays the best performance in terms of PSNR value and payload size. In Fig. 8(c), the scheme of Tai and others [13] generated a payload size above 0.6 bpp, which was better than that generated by the other two methods using the Baboon image. However, the proposed method showed optimal PSNR and a payload size below 0.6 bpp. Figure 8(d) illustrates a similar comparison. Notice that the proposed method could achieve a maximum PSNR with a value exceeding 59 dB, whereas the maximum PSNR for the other two techniques was less than 49 dB.

We also compare our method with three other elegant lossless data hiding methods. Table 3 shows the performance comparison between Awrangjeb and Kankanhalli's algorithm

Table 1. PSNR/payload (dB/bits) generated by using proposed method with various combination of parameters in several images.

Images	β/γ (k=1)			
	1/1	2/1	7/6	12/11
Lena	58.49/8,453	48.48/52,531	37.88/184,052	33.77/208,246
Jet	56.39/13,472	48.26/77,751	38.68/182,165	34.98/202,496
Peppers	59.81/6,172	48.54/39,353	37.76/182,368	32.90/210,262
Baboon	64.10/2,456	48.62/14,214	41.12/86.625	34.40/133,787
Elaine	61.59/5,317	48.59/5,461	39.26/143,269	32.48/204,658
Sailboat	60.74/5,012	48.56/31,325	38.92/147,839	33.20/185,668
GoldHill	60.57/5,176	48.56/32,114	38.95/148,416	33.10/194,038
Man	58.82/9,809	48.49/44,308	39.40/142,899	34.04/180,410
Fishingb.+	60.95/5,101	48.57/29,391	38.96/145,726	33.08/188,764
Tiffany*	57.94/12,342	48.54/53,928	37.99/183,526	33.56/209,538
Average	59.94/7,331	48.52/40,038	38.89/146,035	33.55/191,787

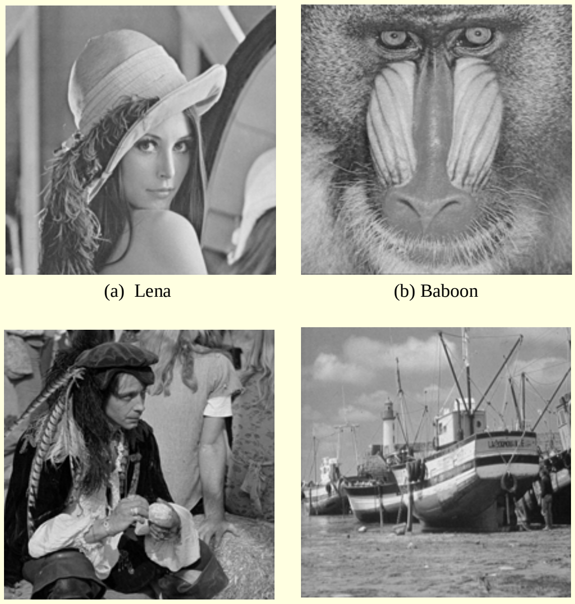
+2 host blocks were skipped with ϕ_1 = 254.
*Maximum preserved algorithm was employed in the image.

Table 2. PSNR/payload (dB/bits) generated via proposed method by using k=2 in several images.

Images	β/γ (k=2)		
	4/3	5/4	6/5
Lena	30.93/271,894	29.46/318,070	28.19/347,864
Jet	31.13/298,512	29.71/329,584	28.53/349,648
Peppers	30.75/236,330	29.35/295,188	28.13/336,238
Baboon	25.05/196,870	24.19/217,610	23.41/236,112
Elaine*	N/A†	N/A	27.72/251,940
Sailboat	N/A	28.97/224,492[a]	27.76/263,476[b]
GoldHill	N/A	28.96/225,430	27.75/264,398
Man	N/A	29.09/231,602[c]	27.86/261,428[c]
Fishingboat	N/A	28.94/218,126[d]	27.73/258,368[d]
Tiffany*	30.99/261,766	29.59/312,268	28.34/345,080
Average	29.77/253,074	28.70/263,597	27.54/291,455

*The maximum preserved approach was employed in the images.
†The case was not available for the image.
[a]3 host blocks were skipped with ϕ_1=241.
[b]24 host blocks were skipped with ϕ_1=238.
[c]2 host blocks were skipped with ϕ_1=246.
[d]63 host blocks were skipped with ϕ_1=239.

[22], [23] and the proposed method. It is obvious that the PSNR of the proposed method outperformed that of Awrangjeb and Kankanhalli's algorithm [22]. At a PSNR value of approximately 38 dB, the payload size generated by the proposed method was seven times larger than that generated by Awrangjeb and Kankanhalli's algorithm. Similarly, the hiding



(a) Lena (b) Baboon (c) Man (d) Fishingboat

Fig. 7. Marked images generated by proposed method: (a) 30.93 dB/1.037 bpp, (b) 29.03 dB/0.664 bpp, (c) 29.09 dB/0.883 bpp, and (d) 28.94 dB/0.832 bpp.

capacity generated by our method was approximately two times that of Awrangjeb and Kankanhalli's algorithm [23]. The PSNR of the proposed method was also superior. Secondly, although the embedding rate generated by Yang and others' scheme [15], 1.120 bpp with a PSNR value of approximately 30 dB, was slightly higher than that of the proposed method (0.965 bpp/29.77 dB), the PSNR of the proposed method was better than that of Yang and others' method when the embedding rate was below 0.5 bpp. For example, the PSNR of the proposed method was 47.25 dB at an embedding rate of 0.279 bpp, while the PSNR of Yang and others' approach was 45.93 dB but at an embedding rate of 0.235 bpp. Moreover, to achieve reversible data hiding with the optimal bit rate of 1.120 bpp, the method of Yang and others must be performed recursively 26 times (with regard to bit embedding and extraction procedures). By contrast, the proposed method requires only a single pass of data embedment and extraction to achieve a similar objective. Therefore, it is not feasible to employ Yang and others' scheme [15] in real-time applications such as lossless data hiding using audio or video as the host media.

IV. Conclusion

This paper proposed a lossless data hiding method based on the MMPOUA algorithm. The first step of the MMPOUA algorithm keeps the minimum pixel (or maximum) of a host

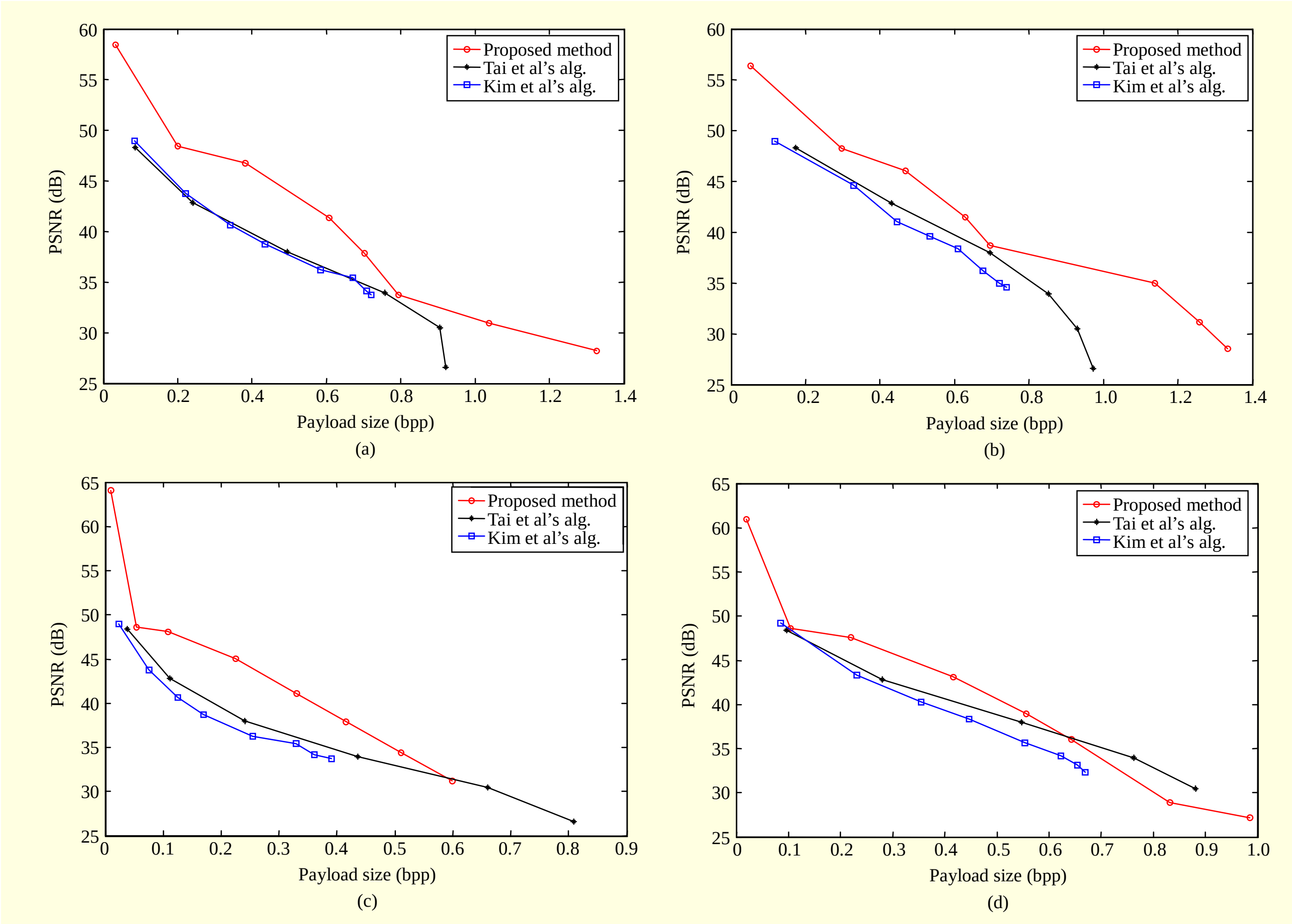


Fig. 8. Performance comparison between the proposed method and existing reversible data hiding schemes in several test images: (a) Lena, (b) Jet, (c) Baboon, and (d) Fishingboat.

Table 3. PNSR/payload (dB/bits) comparison between Awrangjeb and Kankanhalli's algorithm [22], [23] and proposed method in several images.

Images	Methods			
	Ref. [22]	Our method	Ref. [23]	Our method
Lena	38.98/ 15,208	37.88/ 184,052	58.30/ 16,376	48.48/ 52,531
Jet	39.38/ 29,248	38.68/ 182,165	54.64/ 52,520	48.26/ 77,751
Peppers	38.70/ 25,744	37.76/ 182,368	25.75/ 7,288	48.54/ 39,353
Sailboat	36.56/ 10,088	38.92/ 147,839	43.70/ 14,088	48.56/ 31,325
Tiffany	36.38/ 40,696	37.99/ 183,526	51.16/ 15,248	48.54/ 53,928
Average	38.00/ 24,197	38.25/ 175,990	46.71/ 21,104	48.52/ 40,038

block unchanged. The remaining pixels in the block were then subtracted from the minimum (or maximum) pixel. Following pixel squeezing and isolation, data bits were embedded in the reduced (or shifted) block. The proposed method can avoid (or significantly reduce) the use of overhead bits to solve overflow/underflow issues. Experimental results confirmed that the proposed method generated good hiding capacity with high perceived quality, especially at a moderate rate of embedding.

Acknowledgments

The authors would like to thank the editors and anonymous reviewers for providing valuable comments which helped to improve the content of the paper.

References

[1] F.Y. Shih, *Digital Watermarking and Steganography: Fundamentals and Techniques*, FL, USA: CRC Press, 2008.

[2] I.J. Cox et al., *Digital Watermarking and Steganography*, 2nd ed., MA, USA: Morgan Kaufmann, 2008.

[3] W.J. Chen, C.C. Chang, and T.H.N. Le, "High Payload

Steganography Mechanism Using Hybrid Edge Detector," *Expert Syst. Appl*., vol. 37, no. 4, Apr. 2010, pp. 3292-3301.
[4] Z.G. Qu et al., "Novel Quantum Steganography with Large Payload," *Optics Commun*., vol. 283, no. 23, 2010, pp. 4782-4786.
[5] Y.T. Pai and S.J. Ruan, "A High Quality Robust Digital Qatermarking by Smart Distribution Technique and Effective Embedded Scheme," *IEICE Trans. Fundamentals*, vol. E90-A, no. 3, 2007, pp. 597-605.
[6] C.Y. Yang et al., "A Simple Digital Watermarking by the Adaptive Bit-Labeling Scheme," *Int. J. Innovative Computing, Inf. Control*, vol. 6, no. 3, 2010, pp. 1401-1410.
[7] J. Tian, "Reversible Data Embedding Using a Difference Expansion," *IEEE Trans. Circuits Syst. Video Technol*., vol. 13, no. 8, 2003, pp. 890-896.
[8] A.M. Alattar, "Reversible Watermark Using the Difference Expansion of a Generalized Integer Transform," *IEEE Trans. Image Process*., vol. 13, no. 8, 2004, pp. 1147-1156.
[9] S. Weng et al., "Reversible Watermarking Based on Invariability and Adjustment on Pixels Pairs," *IEEE Signal Process. Lett*., vol. 15, 2008, pp. 721-724.
[10] Y. Hu, H.K. Lee, and J. Li, "DE-Based Reversible Data Hiding with Improved Overflow Location Map," *IEEE Trans. Circuits Syst. Video Technol*., vol. 19, no. 2, 2009, pp. 250-260.
[11] Z. Ni et al., "Reversible Data Hiding," *IEEE Trans. Circuits Syst. Video Technol*., vol. 16, no. 3, 2006, pp. 354-362.
[12] C.C. Lin, W.L. Tai, and C.C. Chang, "Multilevel Reversible Data Hiding Based on Histogram Modification of Difference Images," *Pattern Recognition*, vol. 41, no. 12, 2008, pp. 3582-3591.
[13] W.L. Tai, C.M. Yeh, and C.C. Chang, "Reversible Data Hiding Based on Histogram Modification of Pixel Differences," *IEEE Trans. Circuits Syst. Video Technol*., vol. 19, no. 6, 2009, pp. 906-910.
[14] K.S. Kim et al., "Reversible Data Hiding Exploiting Spatial Correlation between Sub-Sampled Images," *Pattern Recognition*, vol. 42, no. 11, 2009, pp. 3083-3096.
[15] H.W. Yang, K.F. Hwang, and S.S. Chou, "Interleaving Max-Min Difference Histogram Shifting Data Hiding Method," *J. Software*, vol. 5, no. 6, 2010, pp. 615-621.
[16] D.M. Thodi and J. Rodriguez, "Expansion Embedding Techniques for Reversible Watermarking," *IEEE Trans. Image Process*., vol. 16, no. 3, 2007, pp. 721-730.
[17] P. Tsai, Y.C. Hu, and H.L. Yeh, "Reversible Image Hiding Scheme Using Predictive Coding and Histogram Shifting," *Signal Process*., vol. 89, no. 6, 2009, pp. 1129-1143.
[18] C.H. Yang and M.H. Tsai, "Improving Histogram-Based Reversible Data Hiding by Interleaving Predictors," *IET Image Process*., vol. 4, no. 4, 2010, pp. 223-234.
[19] C.F. Lee, H.L. Chen, and H.K. Tso, "Embedding Capacity Raising in Reversible Data Hiding Based on Prediction of Different Expansion," *J. Syst. Software*, vol. 83, no. 10, 2010, pp. 1864-1872.
[20] Z. Ni et al., "Robust Lossless Image Data Hiding Designed for Semi-Fragile Image Authentication," *IEEE Trans. Circuits Syst. Video Technol*., vol. 18, no. 4, 2008, pp. 497-509.
[21] X.T. Zeng, L.D. Ping, and X.Z. Pan, "A Lossless Data Hiding Scheme," *Pattern Recognition*, vol. 43, no. 4, 2010, pp. 1656-1667.
[22] M. Awrangjeb and M.S. Kankanhalli, "Lossless Watermarking Considering the Human Visual System," *Lecture Notes Comput. Sci*., vol. 2939, 2004, pp. 329-336.
[23] M. Awrangjeb and M.S. Kankanhalli, "Reversible Watermarking Using a Perceptual Model," *J. Electron. Imaging*, vol. 14, no. 1, 2005, pp. 013014-1-8.
[24] P.G. Howard et al., "The Emerging JBIG2 Standard," *IEEE Trans. Circuits Syst. Video Technol*., vol. 8, no. 7, 1998, pp. 838-848.

Ching-Yu Yang received his BS in electronic engineering in 1983 from National Taiwan Institute of Technology and MS in electrical engineering in 1990 from National Cheng Kung University, Taiwan. In 1999, he received his PhD in computer and information science from National Chiao Tung University. From 1999 to 2005, he was a senior engineer at Chunghwa Telecom. Co. Ltd., Taiwan. He joined the Computer Science and Information Engineering Department, National Penghu University of Technology, in February 2005, and is currently an associate professor there. His recent research interests include image processing, pattern recognition, and data hiding.

Wu-Chih Hu received his PhD in electrical engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, in 1998. Since 1998, he has worked at National Penghu University of Science and Technology. He is currently the chairman and associate professor in the Department of Computer Science and Information Engineering at the same university. Since 1998, he has published more than 70 papers in journals and conference proceedings. His current research interests include computer vision, image processing, pattern recognition, digital watermarking, visual surveillance, and video processing.