Histogram-based Reversible Data Hiding Based on Pixel Differences with Prediction and Sorting

Ya-Fen Chang[1] and Wei-Liang Tai[2]
[1] Department of Computer Science and Information Engineering
National Taichung University of Science and Technology, Taichung, Taiwan
[2] Department of Information Communications
Chinese Culture University, Taipei, Taiwan
[e-mail: cyf@nutc.edu.tw, tai.wei.liang@gmail.com]
*Corresponding author: Wei-Liang Tai

Received September 19, 2011; revised September 18, 2012; accepted November 13, 2012; published December 27, 2012

Abstract

Reversible data hiding enables the embedding of messages in a host image without any loss of host content, which is proposed for image authentication that if the watermarked image is deemed authentic, we can revert it to the exact copy of the original image before the embedding occurred. In this paper, we present an improved histogram-based reversible data hiding scheme based on prediction and sorting. A rhombus prediction is employed to explore the prediction for histogram-based embedding. Sorting the prediction has a good influence on increasing the embedding capacity. Characteristics of the pixel difference are used to achieve large hiding capacity while keeping low distortion. The proposed scheme exploits a two-stage embedding strategy to solve the problem about communicating peak points. We also present a histogram shifting technique to prevent overflow and underflow. Performance comparisons with other existing reversible data hiding schemes are provided to demonstrate the superiority of the proposed scheme.

Keywords: Reversible data hiding, image authentication, lossless watermarking, data hiding, fragile watermarking

This research was supported in part under grant numbers NSC 99-2410-H-025-010-MY2, NSC 101-2410-H-025-009-MY2, and NSC 101-2218-E-034-002-MY2 from the National Science Council, Taiwan.

http://dx.doi.org/10.3837/tiis.2012.12.004

1. Introduction

Hiding data inevitably destroys the host image even though the distortion introduced by hiding is imperceptible to the human visual system [1]. However, there are some sensitive images where any embedding distortion made to the image is intolerable, such as military images, medical images or artwork preservation. For example, even slight changes are not accepted in medical images due to a potential risk of a physician giving a wrong explanation of the image. Hence, reversible data hiding techniques give a solution to the problem of how to embed a large message in digital images in a lossless manner so that the image can be completely restored to its original state before the embedding occurred.

A block diagram of the general reversible data hiding procedure is given in **Fig. 1**. The sender embeds the message *M* to a host image *H* in a lossless manner so that after the message is extracted from the watermarked image, the exact copy of the original image is obtained. Note that even though the distortion introduced by hiding is completely reversible, we are most concerned to minimize the amount of the embedding distortion.

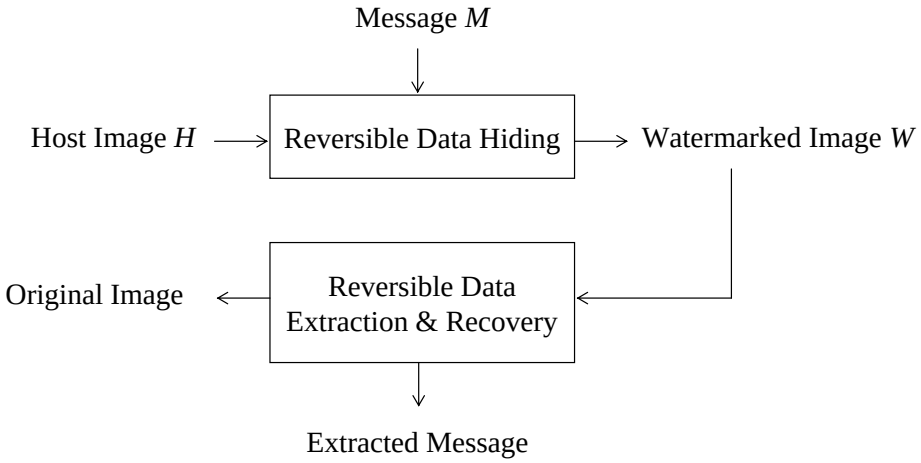


Fig. 1. Diagram for the reversible data hiding procedure

From the application point of view, reversible data hiding technique can be used as a fragile invertible authentication watermarking that embeds an authentication code in a digital image in a reversible way. Only when the embedded authentication code matches the extracted message, the image is deemed authentic. An authenticated party could completely remove the embedding distortion and restore the image to its primitive form. By embedding an authentication code that has a close relationship to the host image, reversible data hiding provides a self authentication scheme without any extra support.

Reversible data hiding techniques have also been proposed for various fields such as audio [2], MPEG-2 video [3], 3D-meshes [4], visible watermarking [5], SMVQ-based compressed domain [6]. In general, the existing reversible data hiding schemes can be classified into three categories: lossless compression, difference expansion (DE), and histogram shifting, among which the histogram-based reversible data hiding schemes have found wide applications for its high watermarked image quality. For lossless compression, Fridrich *et al*. [1] devised an invertible watermarking method by using a lossless compression algorithm to make space in

which to embed data. Later, a lossless generalized LSB embedding scheme (G-LSB) presented by Celik et al. [7][8] uses a variant of an arithmetic compression algorithm to encode a message and hide the resulting interval number in the host image. Although the reversible data hiding schemes based on lossless compression achieve high watermarked image quality, their relative payload is very low.

In 2003, Tian [9] devised a high capacity reversible data hiding technique that is called difference expansion (DE), where the message is embedded based on the 1-D Haar wavelet transform. The resulting high-pass bands are the differences of the adjacent pixel values. The DE scheme is able to embedding as high as 0.15 to 1.97 bpp, which is significantly larger than other schemes proposed previously. Kamstra *et al*. [11] improved the DE scheme by using the low-pass image to find suitable expandable differences in the high-pass band. Thodi and Rodr ́guez [12] proposed an improved version of the DE scheme called prediction-error expansion, where the correlation inherent in the neighborhood of a pixel is better exploited than the DE scheme. Hu *et al*. [13] construct a payload dependent location map, where the compressibility of location map is further improved. Besides, several DE-based schemes [14][15][16] are also proposed recently in which they in principle differ in the employed prediction algorithm. Consequently, DE-based schemes can achieve high embedding capacity but low image quality.

Histogram-based reversible data hiding scheme was first proposed by Ni *et al*. [17] in 2006, where the message is embedded into the histogram bin. They used pairs of peak points and zero points to achieve low embedding distortion but low hiding capacity. Ni et al. [18][19] increased the hiding capacity by extending the histogram modification technique for integer wavelet transform. In 2009, Tsai *et al*. [20] used a residue image indicating a difference between a basic pixel and each pixel in a non-overlapping block to increase the embedding capacity. Tai *et al*. [21] designed a synchronization mechanism by selecting fixed peak bins in histogram of pixel differences. In 2011, Jung *et al*. [22] proposed an improved histogram modification based reversible data hiding technique with a consideration of the human visual system (HVS) characteristics. Yang and Tsai [23] used an interleaving prediction to improve histogram-based reversible data hiding. In 2013, Al-Qershi and Khoo [24] adopt a two-dimensional difference expansion technique (2D-DE) to increase the hiding capacity. Huang and Chang [25] employed modification of difference values between pixels by using histogram-based scheme with extensions to pyramidal structure by utilizing inherent characteristics of original images. Besides, Lou *et al*. [26][27] presented an innovative active steganalysis algorithm for reversible data hiding schemes based on histogram shifting. Their proposed active steganalysis algorithm can effectively detect stego images at low bit rates and estimate the hidden messages locations. Note that histogram-based reversible data hiding schemes can achieve high embedding capacity and high image quality. However, those techniques all suffer from the unresolved issue represented by the need to communicate pairs of peak and zero points to recipients.

In this paper, we focus on improving our previous work [21]. We present an improved histogram-based reversible data hiding scheme based on prediction and sorting. A rhombus prediction is employed to explore the prediction for histogram-based embedding. Sorting the prediction has a good influence on increasing the embedding capacity. Characteristics of the pixel difference are used to achieve large hiding capacity while keeping low distortion. In addition, we exploit a two-stage embedding strategy to solve the problem about communicating peak points. We also present a histogram shifting technique to prevent overflow and underflow.

To make this paper self-contained, in Section 2, we briefly describe prior relevant Ni *et al.*'s histogram modification technique and discuss their limitations. Section 3 contains a detailed exposition of the proposed algorithm. Estimates of distortion are also provided in the same section. In Section 4, we experimentally investigate the relationship between the capacity and distortion, and the influence of variant images on the capacity. We also give the performance comparison with existing reversible schemes in the same section. Finally, the paper is concluded in Section 5, where we outline the future research directions.

2. Histogram Modification

In [17], Ni *et al*. introduced a reversible data hiding scheme based on histogram modification that we will describe briefly in this section. The histogram modification technique involves generating histogram and finding the peak point and the zero point and shifting histogram bins to embed message bits.

For a given host image, we first generate its histogram and find a peak point and a zero point. A peak point corresponds to the grayscale value which the maximum number of pixels in the given image assumes. On the contrary, a zero point corresponds to the grayscale value which no pixel in the given image assumes. For example, the histogram of the grayscale Lena image (512×512×8) is illustrated in **Fig. 2**, in which the peak point is at 154 and the zero point is at 255. Let P be the value of peak point and Z be the value of zero point. The range of the histogram, [P+1, Z-1], is shifted to the right-hand side by 1 to leave the zero point at P+1. Once a pixel with value P is encountered, if the message bit is "1," increase the pixel value by 1. Otherwise, no modification is needed. We note that the number of message bits that can be embedded into an image equals to the number of pixels which are associated with the peak point.

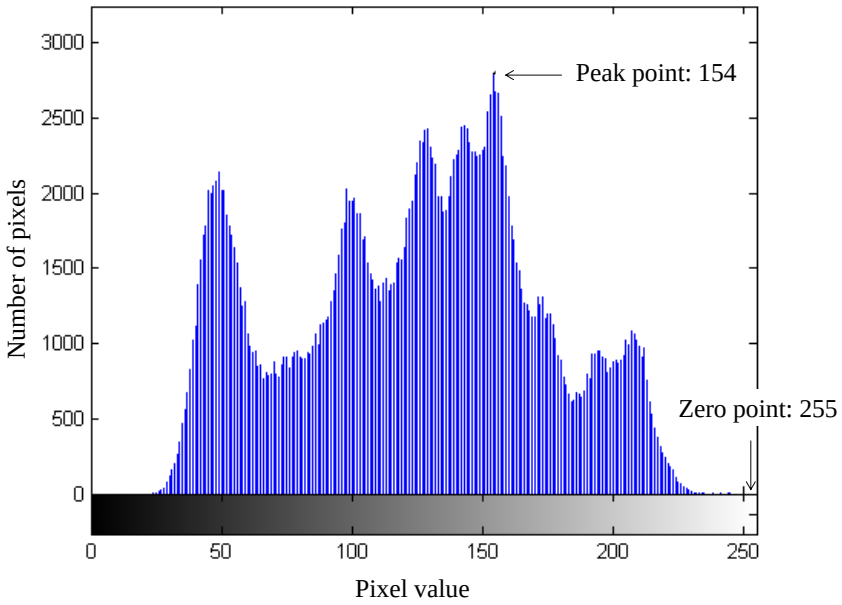


Fig. 2. Histogram of the 'Lena' image

The data extraction is actually the reverse process of data hiding. When a pixel with value P+1 is met, message bit "1" is extracted and the pixel value reduces to P. When a pixel with value

P is met, message bit "0" is extracted. After all message bits have been extracted, shift the range of the histogram, [*P*+2, *Z*], to the left-hand side by 1. Note that zero point defined above may not exist in some image histograms. In this regard, a minimum point that is defined as the grayscale value which the minimum number of pixels in the given image assumes is often used in place of the zero point. However, the grayscale value and coordinate of the pixel that is associated with the minimum point need to be recorded as overhead bookkeeping information. Therefore, if the required payload is larger than the actual hiding capacity that is referred to as pure payload, more pairs of peak and minimum points need to be used.

3. Proposed Scheme

We note that the histogram-based reversible data hiding schemes do not work well in a case where an image having an equal histogram. The pure payload is still a little low even we use multiple pairs of peak and zero points for embedding. What is more, the histogram modification technique has an unsolved issue that multiple pairs of peak and minimum points have to be communicated to the recipient via a side channel.

Thus, we present an efficient extension of the histogram modification technique by considering the difference between adjacent pixels instead of simple pixel value. The distribution of pixel difference has a prominent maximum since image neighbor pixels are strongly correlated. Hence, there are a lot of candidates for embedding data, as shown in **Fig. 3**. Characteristics of the pixel difference that is Laplacian-like distributed are exploited to achieve large hiding capacity while keeping the embedding distortion low. This observation leads us toward designs in which the embedding is done in pixel differences. Meantime, we find that sorting the prediction has much shaper histogram which would lead to significant performance improvement for histogram-based embedding. As a result, a rhombus prediction is employed in our scheme for increasing the embedding capacity. We also use a histogram shifting technique to prevent overflow and underflow. Furthermore, we use a two-stage embedding strategy to solve the problem about communicating peak points. In the following, we now outline the principle of the proposed reversible data hiding algorithm.

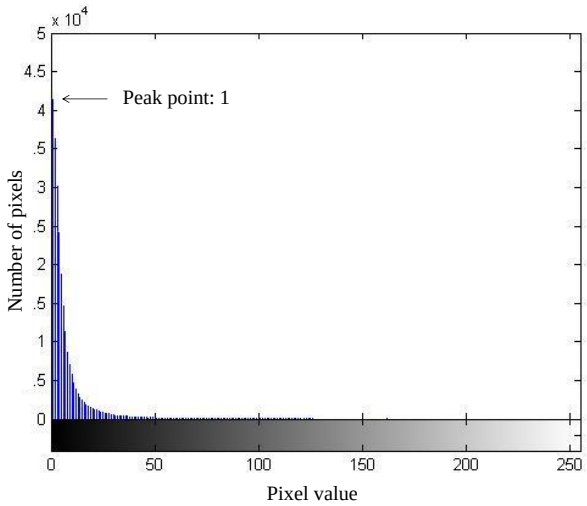


Fig. 3. The difference image histogram

3.1 Rhombus Prediction and Sorting

Our previous work's [21] use of pixel difference histogram introduced a significant performance advantage over previous methods. To improve our previous work, we present the prediction sorting to enhance the correlation of neighboring pixels. In order to predict the pixel value of position $u_{i,j}$ in **Fig. 4**, we use a rhombus prediction by considering four neighboring pixels $v_{i,j-1}$, $v_{i-1,j}$, $v_{i,j+1}$, $v_{i+1,j}$. All pixels of the image are divided into two sets: the "White" set and "Gray" set. The pixel value u of the White set can be predicted by using the four neighboring pixel values of the Gray set and to hide data. Note that the two sets are independent, which means changes in one set do not affect the other set, and vice versa. The center pixel $u_{i,j}$ can be predicted from the four neighboring pixels $v_{i,j-1}$, $v_{i-1,j}$, $v_{i,j+1}$, $v_{i+1,j}$. The predicted value $u'_{i,j}$ is computed as follows:

$$u'_{i,j} = \left\lfloor \frac{v_{i,j-1} + v_{i-1,j} + v_{i,j+1} + v_{i+1,j}}{4} \right\rfloor \tag{1}$$

In order to hide more data with less visual degradation, the order to hide data into the pixel difference needs to be changed. Thus, the cover pixels can be rearranged by sorting according to the prediction of neighboring pixels. To ensure the reversibility, we use a stable sorting algorithm to sort the prediction values. Sorting cover pixels according to the prediction values enables hiding data in pixel difference with high embedding capacity. As a result, this observation leads us toward designs in which the embedding is done in pixel differences according to the prediction sorting. Hiding data according to the sorting adapted to the rhombus prediction will be presented in more detail in Section 3.2.

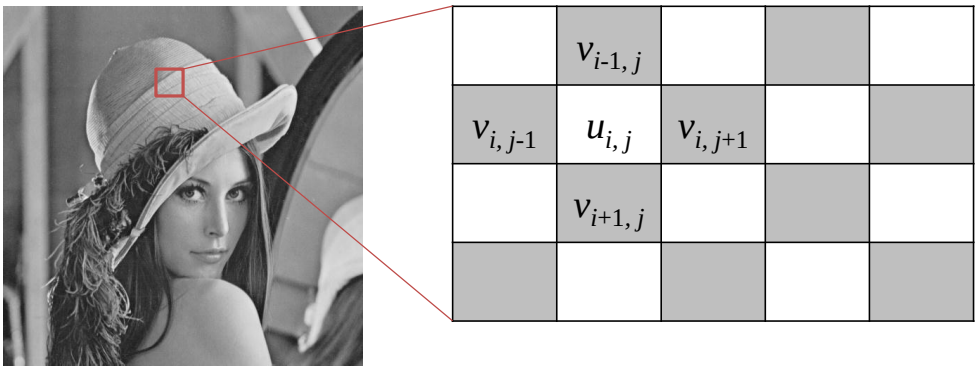


Fig. 4. Rhombus prediction

3.2 Histogram Modification on Pixel Differences

The reversible data hiding scheme for White set is designed as follows.

1) Predict the pixel value $u_{i,j}$ in White set using Eq. 1.
2) Sort the host pixel $u_{i,j}$ according to the prediction value $u'_{i,j}$, and produce the sorted pixels $\{x_0, x_1, \ldots, x_i\}$ for $0 \le i \le N\text{-}1$ where N is the pixel number of White set.
3) Calculate the pixel difference d_i between pixels x_{i-1} and x_i by

$$d_i = \begin{cases} x_i, & \text{if } i = 0, \\ |x_{i-1} - x_i|, & \text{otherwise.} \end{cases} \tag{2}$$

4) Determine the peak point P from the pixel differences.

5) If $d_i > P$, shift x_i by 1 unit:

$$y_i = \begin{cases} x_i, & \text{if } i = 0 \text{ or } d_i < P, \\ x_i + 1, & \text{if } d_i > P \text{ and } x_i \geq x_{i-1}, \\ x_i - 1, & \text{if } d_i > P \text{ and } x_i < x_{i-1}, \end{cases} \tag{3}$$

where y_i is the watermarked value of pixel i.

6) If $d_i = P$, modify x_i according to the message bit:

$$y_i = \begin{cases} x_i + b, & \text{if } d_i = P \text{ and } x_i \geq x_{i-1}, \\ x_i - b, & \text{if } d_i = P \text{ and } x_i < x_{i-1}, \end{cases} \tag{4}$$

where y_i is the watermarked value of pixel i.

7) Construct the watermarked White set according to the sorted pixels $\{y_0, y_1, \ldots, y_i\}$ for $0 \leq i \leq N$-1 where N is the pixel number of White set.

The embedding scheme for White set computes predicted values using the Gray set and embeds data using the White set. Thus, the output of the embedding scheme for White set is the unchanged pixels from the Gray set and the watermarked pixels from the White set. Similarly, we can embed data in Gray set by considering the predicted values using the watermarked White set. The White and Gray embedding schemes are similar in nature. As a result, the consecutive usage of the White embedding scheme and the Gray embedding scheme results in nearly double the embedding capacity.

At the receiving end, the recipient extracts message bits from the watermarked image by scanning the image in the same order as during the embedding. The recipient sorts the watermarked pixel $u_{i,j}$ in White set according to the prediction value $u'_{i,j}$, and produces the sorted pixels $\{y_0, y_1, \ldots, y_i\}$. The message bit b can be extracted by

$$b = \begin{cases} 0, & \text{if } |y_i - x_{i-1}| = P, \\ 1, & \text{if } |y_i - x_{i-1}| = P + 1, \end{cases} \tag{5}$$

where x_{i-1} denotes the restored value of y_{i-1}. Then the original pixel value of x_i can be restored by

$$x_i = \begin{cases} y_i + 1, & \text{if } |y_i - x_{i-1}| > P \text{ and } y_i < x_{i-1}, \\ y_i - 1, & \text{if } |y_i - x_{i-1}| > P \text{ and } y_i > x_{i-1}, \\ y_i, & \text{otherwise.} \end{cases} \tag{6}$$

Thus, the exact copy of the original host image is obtained.

Fig. 5 shows an embedding example for White set of a grayscale image with 4×4 pixels. We first predict the pixel value $u_{i,j}$ in White set using Eq. 1. Sort the host pixel $u_{i,j}$ according to the prediction value $u'_{i,j}$, and produce the sorted pixels $\{x_0, x_1, \ldots, x_i\}$. We then calculate the pixel difference d_i between pixels x_{i-1} and x_i. Thus, the peak point is 0 and the corresponding number is 5. Let us assume that the message bit-stream to be embedded is 01101. Since $|x_0-x_1|=|155-155|=0=P$, the first message bit 0 is embedded in x_1 by leaving x_1 unmodified. The difference between x_1 and x_2 is $|155-155|=0=P$, then the second message bit 1 is embedded in x_2 by setting $y_2 = x_2 + 1$ since $x_2 > x_1$. As $|x_2-x_3|=|155-156|=2 > P$ and $x_3 > x_2$, $y_3 = x_3 + 1=157$. The embedding process continues until all of message bits are embedded, and then the resulting watermarked pixels are obtained. Finally, we construct the watermarked White set according to the sorted pixels $\{y_0, y_1, \ldots, y_i\}$.

Given P=0, we can completely restore the image to its original state before the embedding occurred. The recipient sorts the watermarked pixel $u_{i,j}$ in White set according to the

prediction value $u'_{i,j}$, and produces the sorted pixels $\{y_0, y_1, \ldots, y_i\}$. Since $|y_1-x_0| = |155-155| = 0 = P$, a message bit 0 is extracted and $x_1 = y_1$. The difference between y_2 and x_1 is $|156-155|=1=P+1$, a message bit 1 is extracted and x_2 is restored by setting $x_2 = y_2 - 1=155$ since $y_2 > x_1$. Since $|y_3-x_2|=|157-155|=2 > P$ and $y_3 > x_2$, x_3 is restored by setting $x_3 = y_3 - 1=156$. The extraction process continues until all of message bits are extracted. Thus, the watermarked image is reverted to the exact copy of the original host image.

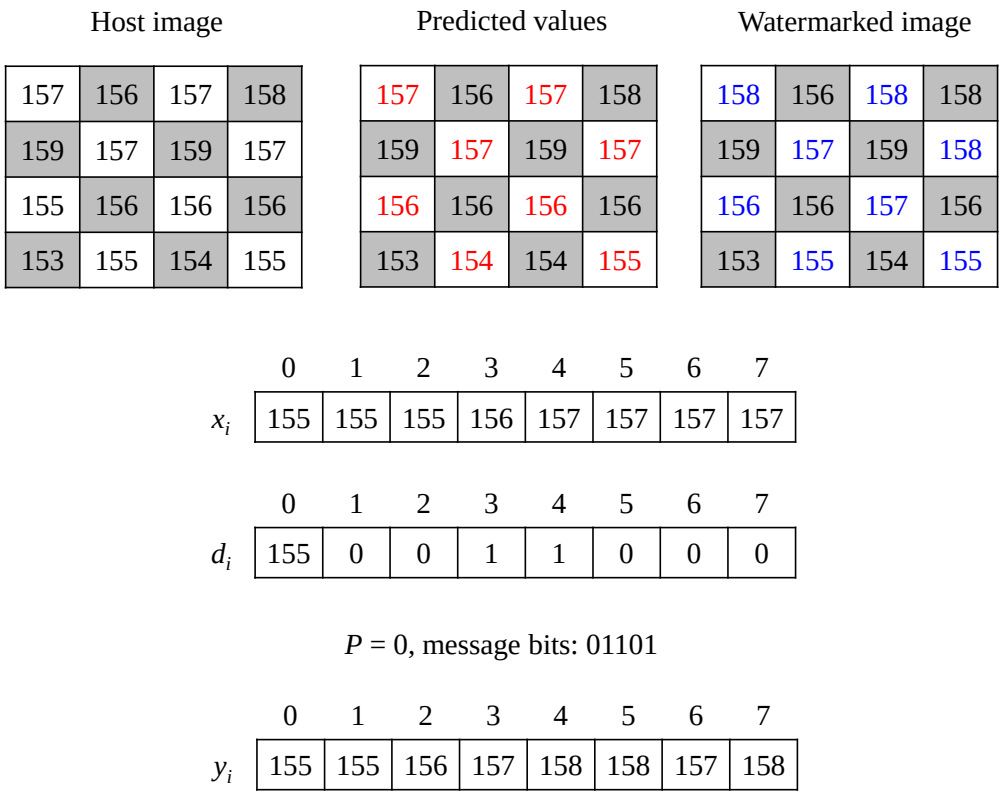


Fig. 5. An example of the reversible data hiding scheme for White set

The above steps complete the data hiding process where only the White set is used to embed data. The White and Gray embedding schemes are similar in nature. Note that large embedding capacities can be obtained by repeated data hiding process in White set and Gray set.

3.3 Prevent Overflow and Underflow

Modification to a pixel may not be allowed if the pixel is saturated (0 or 255). Pixels causing overflow or underflow errors should be excluded. The condition

$$0 \le y_i \le 255$$

is used for finding such problematic pixels. To prevent overflow and underflow, we adopt a histogram shifting technique [21] that narrows the histogram from both sides as shown in **Fig. 6**.

Note that a single layer data hiding process consists of White and Gray data hiding. Let us assume that the number of peak points that we use to embed messages is L when we adopt the proposed L-layer data hiding scheme. Thus, we shift the histogram from both sides by L unit to

prevent overflow and underflow since the pixel x_i that satisfies $d_i \geq L$ will be shifted by L unit after the embedding occurred.

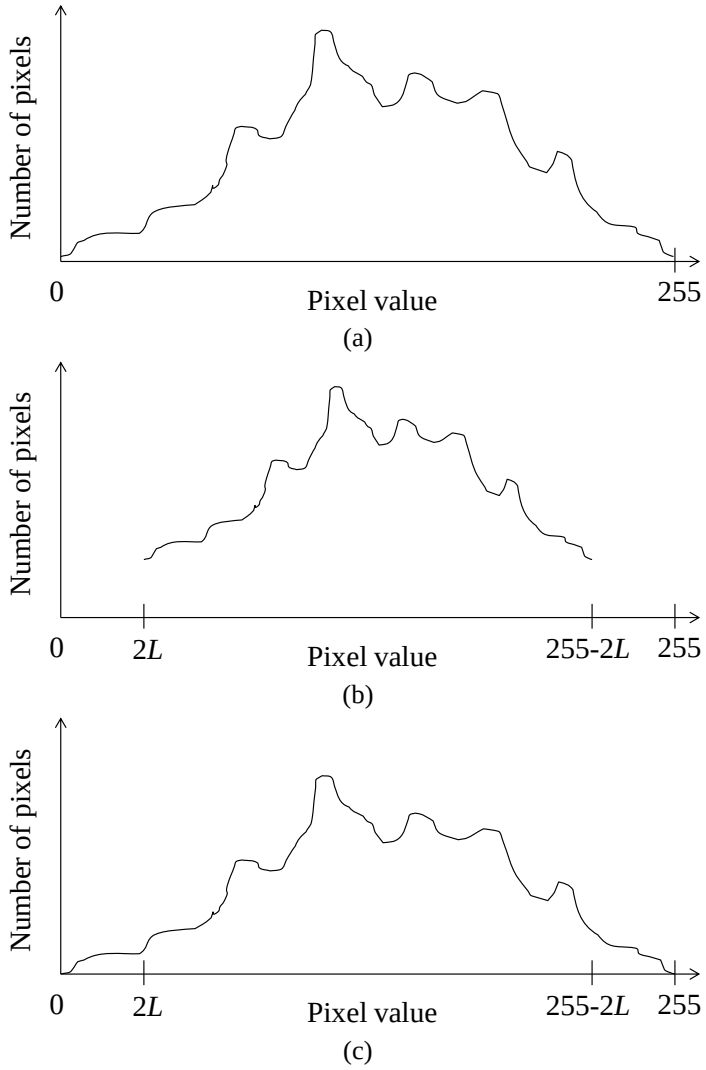


Fig. 6. Histogram shifting; (a) original histogram, (b) histogram shifting, and (c) histogram after data hiding

After narrowing down the histogram to the range [L, 255-L], we need to record the histogram shifting information as overhead bookkeeping information. For this purpose, we create a one-bit map as the location map, whose size is equal to the size of host image. For a pixel whose grayscale value is in the range [L, 255-L], we assign a value 0 in the location map; otherwise, we assign a value 1. The location map is losslessly compressed by the run-length coding algorithm, which enables a large increase in compression ability since pixels out of the range [L, 255-L] are few and almost contiguous. We know that the maximum modification to a pixel is limited to L according to the proposed scheme. As a result, shifting the histogram from both sides by L units enables us to avoid occurring overflow and underflow. Note that peak points at every hiding pass and the location map need to be forwarded to the recipient in order to exactly extract the message. Thus, the overhead information needs to be embedded into the

host image together with the embedded message. Next, we use two-stage embedding strategy to solve the problem about communicating peak points.

3.4 Two-stage Embedding

For recovering data, the side information, namely peak points and the location map, should be sent to the recipient. Assume that the number of peak points that we use to embed messages is $2L$ when we adopt the proposed L-layer data hiding scheme and $|Map|$ is the length of the location map. Thus, when the first (L-1)-layer embedding is finished, we perform the two-stage embedding on the sorted pixels $\{x_0, x_1, \ldots, x_i\}$ to complete the final layer embedding. The LSB values of the first $(2L \times 8)+|Map|$ host pixels from the sorted pixels $\{x_0, x_1, \ldots, x_i\}$ are replaced with peak points ($2L \times 8$ bits) and location map ($|Map|$ bits). Original $(2L \times 8)+|Map|$ LSB values should be collected and included to the payload. As a result, these $(2L \times 8)+|Map|$ host pixels should be excluded from the sorted pixels $\{x_0, x_1, \ldots, x_i\}$. We note that the overhead information is included in the image itself with payload. Thus, the real capacity Cap that is referred to pure payload is

$$Cap = N_p - |O|,$$

where N_p is the number of pixels which are associated with peak points and $|O|$ is the length of the overhead information.

3.5 Lower Bound of PSNR

Obviously, the pixel x_i whose difference d_i is larger than peak point will be either increased or decreased by 1 in the data embedding process with one peak point. Therefore, in the worse case, all pixel values will be increased or decreased by 1 but the first pixel. That is, the resulted the mean squared error (MSE) is (N-1)/N, which is almost equal to 1 when N is large enough. Thus, the lower bound of PSNR for the watermarked image generated from the embedding process with one peak point is

$$\text{PSNR (dB)} = 10 \times \log_{10}\left(\frac{255^2}{\text{MSE}}\right) \geq 48.13 \text{ dB}.$$

As a result, the lower bound of PSNR for the watermarked image generated by our proposed algorithm with one peak point is theoretically proved larger than 48 dB, which is also supported by our numerous experiments.

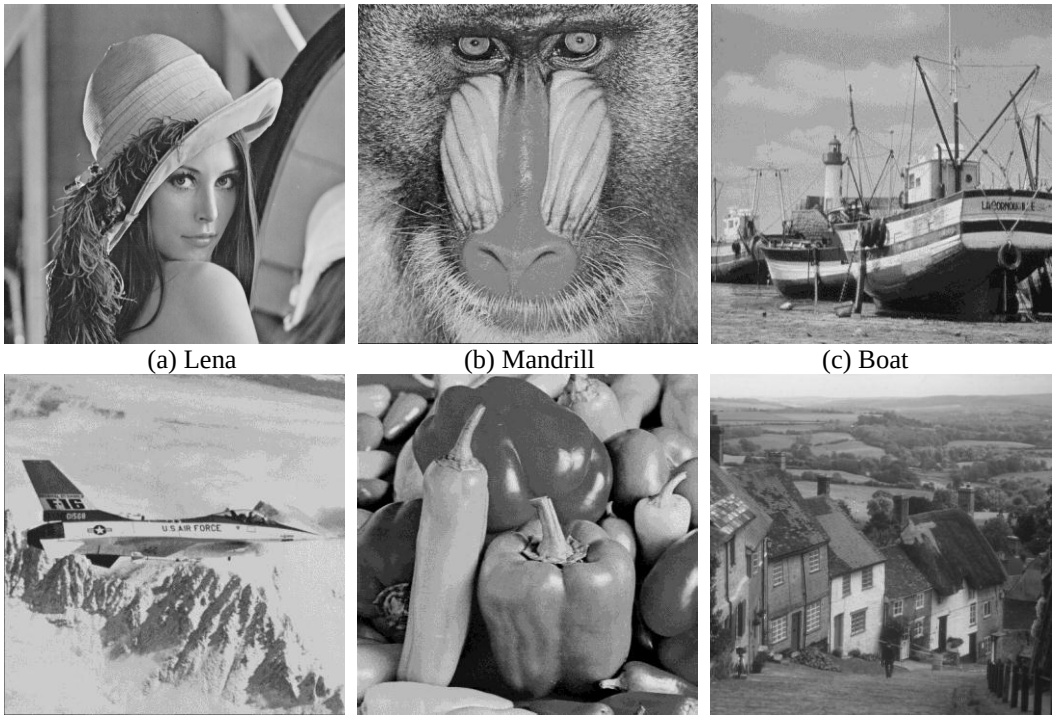
4. Experimental Results

To obtain a better understanding of how various images affect the performance of the proposed reversible data hiding scheme, we present some results in a graphical form. All experiments were performed with six grayscale test images of size 512×512, Lena, Mandrill, Boat, F-16, Peppers, and GoldHill, as shown in **Fig. 7**.

4.1 Capacity versus Distortion Performance

In **Table 1**, we give an example of how different images influence the pure payload Cap and the distortion at the first layer embedding. We can see a very high variability in N_p which is defined as the number of pixels which are associated with peak points, between images. Smoother and less noisy images lead to a larger N_p than images that are highly textured or noisy. We have also observed that there are no pixels without the range [1, 254] except for Mandrill. For Lena, the size of location map is $|MAP|$=20 bits when the run-length coding

algorithm is applied to compress the location map. Thus, the size of overhead information is $|O|=(2L\times 8)+|Map|=36$ bits. Further, the PSNR for the watermarked images shown in Table 1 closely matches our theoretical estimated lower bound.



(a) Lena (b) Mandrill (c) Boat

(d) F-16 (e) Peppers (f) GoldHill

Fig. 7. Six test images used for performance evaluation

Table 1. Hiding capacity and distortion for six test images with L=1

Host image (512×512)	N_p	Pure payload *Cap* (bits)	Overhead information $\|O\|$ (bits)	PSNR (dB)	Bit rate (bpp)
Lena	45074	45038	36	48.98	0.1718
Mandrill	14497	14361	136	48.38	0.0548
Boat	46461	46425	36	49.01	0.1771
F-16	65734	65698	36	49.54	0.2506
Peppers	31321	31285	36	48.69	0.1193
GoldHill	33537	33501	36	48.74	0.1278

Table 2 shows a number of experiments to see how the pure payload and distortion change with multiple layer embedding. We can see that images with abundant highly textured and noisy areas have generally smaller capacity since those images influence the effectiveness of prediction. We also note that the pure payload increases very fast with L. For some images, however, the pure payload abruptly decreases with increased L since the length of overhead information increases very fast with L. In **Table 3**, we give the overhead information (no. of pixels) for multiple layer embedding. Notice that for level L, we record pixels whose values are out of the range [L, 255-L] as overhead information. We see that images with large dark and bright areas have generally large overhead information.

Table 2. Pure payload for test images with L-layer embedding

Host image (512×512)	Pure payload *Cap* for *L*-layer embedding, *L* = 1, 2, …, 6					
	1	2	3	4	5	6
Lena	45038	78202	103668	125578	145867	163001
Mandrill	14361	26969	38707	49033	58829	67768
Boat	46425	80408	106720	128738	148810	166822
F-16	65698	111870	146352	173439	196326	217973
Peppers	31285	55804	75837	94312	109895	124216
GoldHill	33501	59585	80565	100484	117400	132940
Average PSNR (dB)	48.89	43.73	40.71	38.53	36.83	35.41

Table 3. Overhead information (no. of pixels) for *L*-layer embedding

Host image (512×512)	Overhead information (no. of pixels) for *L*-layer embedding = 1, 2, 4,…, 16				
	1	2	4	8	16
Lena	0	0	0	0	4
Mandrill	33	45	92	202	295
Boat	0	0	0	1	23
F-16	0	0	0	0	0
Peppers	0	17	156	1932	9528
GoldHill	0	0	0	0	0

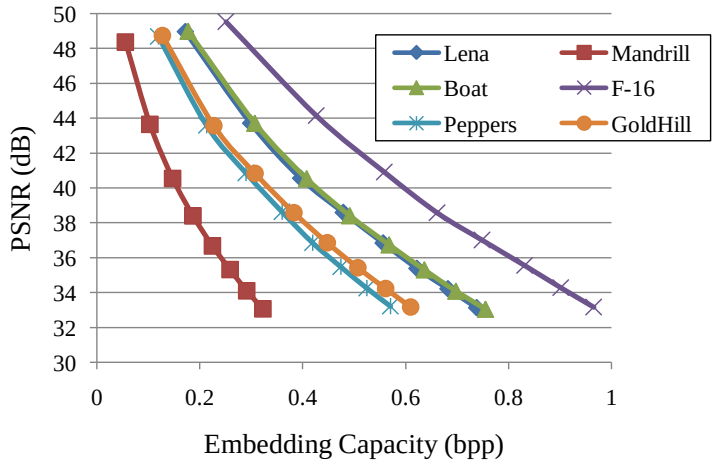


Fig. 8. PSNR versus embedding capacity for test images with 8-layer embedding

The PSNR of the watermarked images is plotted against the embedding capacity for test images with 8-layer embedding in **Fig. 8**. We note that the achievable real capacity depends on the nature of the image itself. Smooth images provide higher real capacity at the same embedding distortion. Therefore, images with high correlation have better performance than images with low correlation.

Figs. 9 and **10** show the visual impacts of watermarked images at various embedding capacities for Lena and Mandrill, respectively. Obviously, the watermarked image hardly can be distinguished from the original image. Although a slight sharpening effect can be attentively observed when the original and the watermarked images are alternately displayed, the highlight phenomenon is not perceptually obvious, even at low PSNR. For the smooth image Lena, while the pure payload size is close to 1 bpp, the visual distortion is still quite small and the PSNR is higher than 30 dB. As a result, images with high textured areas and low correlation produce less N_p than smooth images, and hence, embed less information at lower PSNR.

Fig. 9. Original and watermarked Lena; (a) original, (b) 48.98 dB embedded with 0.1718 bpp, (c) 38.59 dB embedded with 0.479 bpp, and (d) 30.02 dB embedded with 0.918 bpp

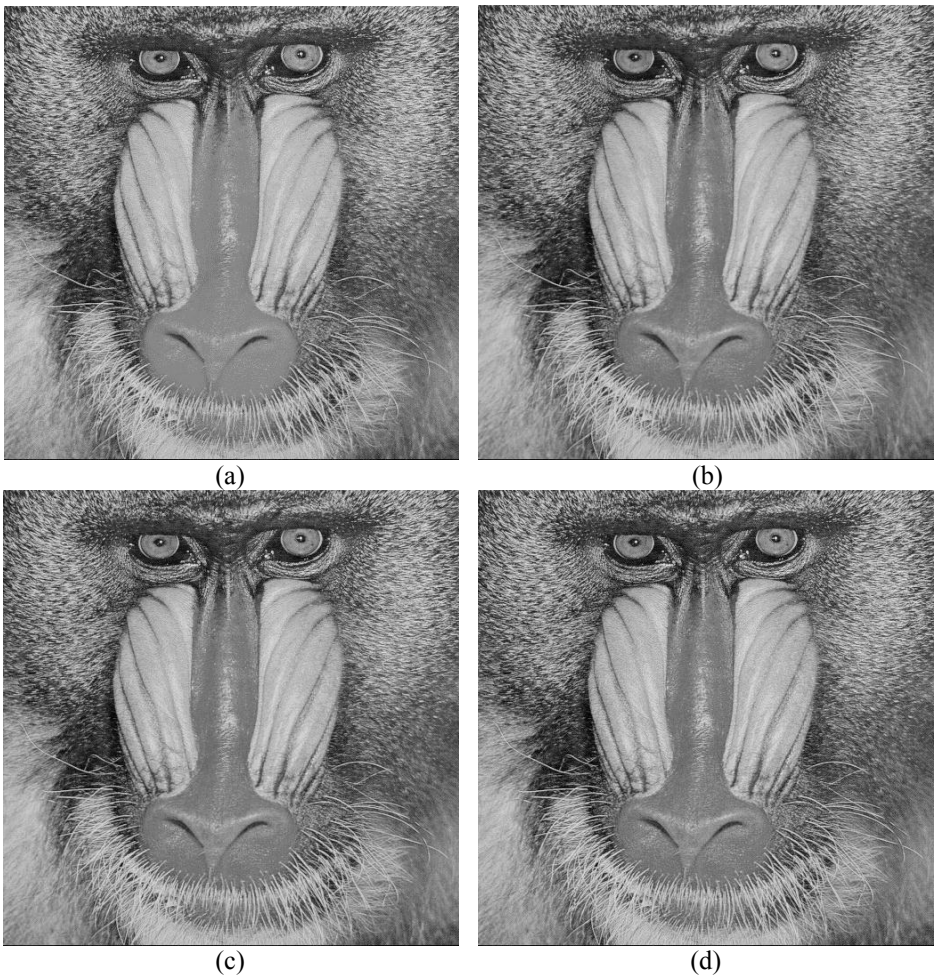


Fig. 10. Original and watermarked Mandrill; (a) original, (b) 48.38 dB embedded with 0.055 bpp, (c) 36.67 dB embedded with 0.224 bpp, and (d) 33.06 dB embedded with 0.323 bpp

4.2 Comparison with Other Recent Schemes

Fig. 11 shows the comparison of embedding capacity in bpp versus image quality in PSNR of the proposed scheme with that of the existing reversible schemes [13][21][22][24][25] for Lena. Note that the proposed scheme and schemes [21][22][25] were proposed for histogram modification, whereas scheme [13][24] was presented for difference expansion. Scheme [25] achieved performance similar to our proposed scheme; however, their algorithms did not provide a solution to the problem of communicating threshold values and multiple pairs of peak and minimum points. The proposed scheme uses prediction and sorting to embed data into pixel differences. Thus, we have improved our previous work [21] and derived better performance. The evaluation results in **Fig. 11** show that the proposed scheme achieves high pure capacity with low distortion. As a result, by use of prediction strategy, it is a good way for obtaining better performance in reversible data hiding.

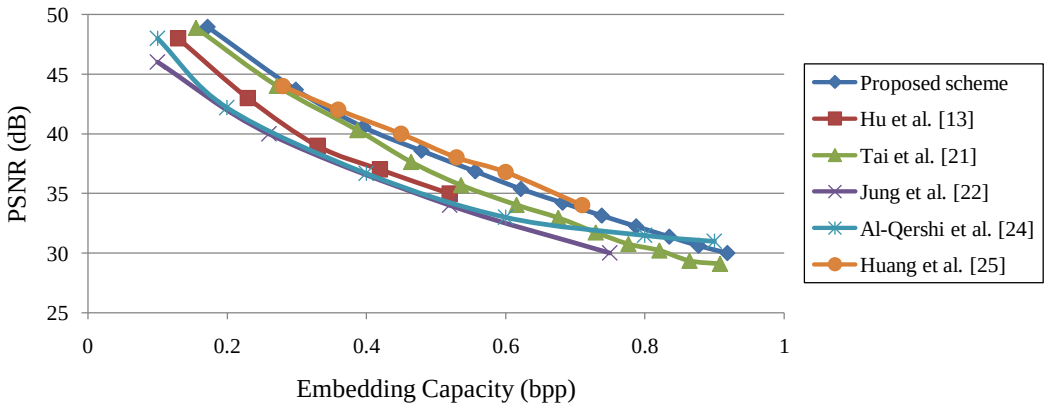


Fig. 11. Performance comparison for Lena with existing reversible schemes [13][21][22][24][25]

5. Conclusion

In this paper, we present an efficient extension of histogram modification by considering the difference between adjacent pixels instead of simple pixel value. The distribution of pixel difference has a prominent maximum since neighbor pixels are strongly correlated. Further, we use prediction and sorting to enhance the correlation of neighbor pixels in order to improve the embedding capacity. In addition, one common problem of virtually histogram-based techniques is that they have to transmit pairs of peak and minimum points to recipients. To solve this problem, we introduce the two-state strategy to embed the overhead information. We also use a histogram shifting technique to prevent overflow and underflow. As a result, the evaluation results show that the proposed scheme have significantly improved our previous work [21] and derived better performance.

The proposed scheme provides high capacities at small and invertible distortion. It can be easily applied for compressed image formats, such as JPEG, MPEG, and JPEG2000, since the distribution of frequency coefficients is almost Laplacian distributed due to quantization and typical characteristics of images. Thus, the proposed scheme is able to be easily performed in the transform domain to improve the hiding ability.

References

[1] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," In *Proc. of the SPIE, Security and Watermarking of Multimedia Contents III*, vol. 3971, San Jose, California, Jan. 2001, pp. 197-208. Article (CrossRef Link).

[2] M. Van der Veen, F. Bruekers, A. Van Leest, and S. Cavin, "High capacity reversible watermarking for audio," In *Proc. of the SPIE, Security and Watermarking of Multimedia Contents V*, vol. 5020, Santa Clara CA, Jan. 2003, pp. 1-11. Article (CrossRef Link).

[3] D. Rui and J. Fridrich, "Lossless authentication of MPEG-2 video," In *Proc. of IEEE International Conference on Image Processing*, vol. 2, Rochester, NY, 2002, pp. 893-896. Article (CrossRef Link).

[4] J. Dittmann and O. Benedens, "Invertible authentication for 3D-meshes," In *Proc. of the SPIE, Security and Watermarking of Multimedia Contents V*, vol. 5020, Santa Clara CA, Jan. 2003, pp. 653-664. Article (CrossRef Link).

[5] Y. Hu and B. Jeon, "Reversible visible watermarking and lossless recovery of original images,"

IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 11, pp. 1423-1429, Nov. 2006. Article (CrossRef Link).
[6] C. C. Chang, W. L. Tai, and C. C. Lin, "A reversible data hiding scheme based on side match vector quantization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 10, pp. 1301-1308, Oct. 2006. Article (CrossRef Link).
[7] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 253-266, Feb. 2005. Article (CrossRef Link).
[8] M. U. Celik, G. Sharma, and A. M. Tekalp, "Lossless watermarking for image authentication: a new framework and an implementation," *IEEE Transactions on Image Processing*, vol. 15, no. 4, pp. 1042-1049, Apr. 2006. Article (CrossRef Link).
[9] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890-896, Aug. 2003. Article (CrossRef Link).
[10] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147-1156, Aug. 2004. Article (CrossRef Link).
[11] L. Kamstra and H. J. A. M. Heijmans, "Reversible data embedding into images using wavelet techniques and sorting," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2082-2090, Dec. 2005. Article (CrossRef Link).
[12] D. M. Thodi and J. J. Rodríguez, "Expansion embedding techniques for reversible watermarking," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 721-730, Mar. 2007. Article (CrossRef Link).
[13] Y. Hu, H. K. Lee, and J. Li, "DE-based reversible data hiding with improved overflow location map," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 2, pp. 250-260, Feb. 2009. Article (CrossRef Link).
[14] D. Coltuc, "Improved embedding for prediction-based reversible watermarking," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 873-882, Sept. 2011. Article (CrossRef Link).
[15] X. Li, B. Yang, and T. Zeng, "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3524-3533, Dec. 2011. Article (CrossRef Link).
[16] D. Coltuc, "Low distortion transform for reversible watermarking," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 412-417, Jan. 2012. Article (CrossRef Link).
[17] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354-362, Mar. 2006. Article (CrossRef Link).
[18] D. Zou, Y. Q. Shi, Z. Ni, and W. Su, "A semi-fragile lossless digital watermarking scheme based on integer wavelet transform," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 10, pp. 1294-1300, Oct. 2006. Article (CrossRef Link).
[19] Z. Ni, Y. Q. Shi, N. Ansari, W. Su, Q. Sun, and X. Lin, "Robust lossless image data hiding designed for semi-fragile image authentication," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 4, pp. 497-509, Apr. 2008. Article (CrossRef Link).
[20] P. Tsai, Y. C. Hu, and H. L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Processing*., vol. 89, pp. 1129-1143, 2009. Article (CrossRef Link).
[21] W. L. Tai, C. M. Yeh, and C. C. Chang, "Reversible data hiding based on histogram modification of pixel differences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 906-910, Jun. 2009. Article (CrossRef Link).
[22] S. W. Jung, L. T. Ha, and S. J. Ko, "A new histogram modification based reversible data hiding algorithm considering the human visual system," *IEEE Signal Processing Letters*, vol. 18, no. 2, pp 95-98, Feb. 2011. Article (CrossRef Link).
[23] C. H. Yang and M. H. Tsai, "Improving histogram-based reversible data hiding by interleaving predictions," *IET Image Processing*, vol. 4, no. 4, pp. 223-234, 2010. Article (CrossRef Link).

[24] O. M. Al-Qershi and B. E. Khoo, "Two-dimensional difference expansion (2D-DE) scheme with a characteristics-based threshold," *Signal Processing*, vol. 93, pp. 154-162, 2013. Article (CrossRef Link).
[25] H. C. Huang and F. C. Chang, "Hierarchy-based reversible data hiding," *Expert Systems with Applications*, vol. 40, pp. 34-43, 2013. Article (CrossRef Link).
[26] D. C. Lou, C. H. Hu, and C. C. Chiu, "Steganalysis of histogram modification reversible data hiding scheme by histogram feature coding," *International Journal of Innovative Computing Information and Control*, vol. 7, no. 11, pp. 6571-6583, Nov. 2011. Article (CrossRef Link).
[27] D. C. Lou, C. L. Chou, H. K. Tso, and C. C. Chiu, "Active steganalysis for histogram-shifting based reversible data hiding," *Optics Communications*, vol. 285, no. 10-11, pp. 2510-2518, Apr. 2012. Article (CrossRef Link).

Ya-Fen Chang is an Associate Professor of Department of Computer Science and Information Engineering at National Taichung University of Science and Technology in Taiwan. She received her BS degree in computer science and information engineering from National Chiao Tung University and Ph.D. degree in computer science and information engineering from National Chung Cheng University, Taiwan. Her current research interests include electronic commerce, information security, cryptography, mobile communications, image processing, and data hiding.

Wei-Liang Tai received the M.S. degrees in computer science and information engineering in 2004 from National Chung Cheng University, Taiwana and the Ph.D. degree in computer science and information engineering from National Chung Cheng University, Taiwan in 2008. He is currently Assistant Professor, Department of Information Communications, Chinese Cluture University, Taiwan, with main research interests on information security and forensics and multimedia signal processing.