Reversible Data Hiding Based on Histogram Modification of Pixel Differences

Wei-Liang Tai, Chia-Ming Yeh, and Chin-Chen Chang, *Fellow, IEEE*

***Abstract*— In this letter, we present a reversible data hiding scheme based on histogram modification. We exploit a binary tree structure to solve the problem of communicating pairs of peak points. Distribution of pixel differences is used to achieve large hiding capacity while keeping the distortion low. We also adopt a histogram shifting technique to prevent overflow and underflow. Performance comparisons with other existing schemes are provided to demonstrate the superiority of the proposed scheme.**

***Index Terms*— Image authentication, lossless watermarking, reversible data hiding.**

I. Introduction

DATA HIDING is a term encompassing a wide range of applications for embedding messages in content [1], [2]. Inevitably, hiding information destroys the host image even though the distortion introduced by hiding is imperceptible to the human visual system. There are, however, some sensitive images for which any embedding distortion of the image is intolerable, such as military images, medical images, or artwork preservation. For medical images, even slight changes are unacceptable because of the potential risk of a physician misinterpreting the image. Consequently, reversible data hiding techniques are designed to solve the problem of lossless embedding of large messages in digital images so that after the embedded message is extracted, the image can be completely restored to its original state before embedding occurred.

Fridrich *et al*. [3], [4] devised an invertible watermarking method by using a lossless compression algorithm to make space in which to embed data. De Vleeschouwer *et al*. [5] proposed a semi-fragile technique based on the patchwork algorithm and modulo operation. A lossless generalized LSB embedding scheme (G-LSB) presented by Celik *et al*. [6], [7] uses a variant of an arithmetic compression algorithm to encode a message and hide the resulting interval number in the host image. Tian [8] devised a high-capacity reversible data hiding technique called difference expansion (DE), in which the message is embedded based on the 1-D Haar wavelet transform. The resulting high-pass bands are the differences between adjacent pixel values. Tian's technique has been extended recently in [9]–[11].

Reversible data hiding techniques have also been proposed for various fields such as audio [12], MPEG-2 video [13], 3-D meshes [14], visible watermarking [15], SMVQ-based compressed domain [16], and the integer-to-integer wavelet domain [17]. Another novel histogram-based reversible data hiding technique was presented by Ni *et al*. in [18], in which the message is embedded into the histogram bin. They used peak and zero points to achieve low distortion, but with attendant low capacity. Histogram modification techniques have been extended recently in [19], [20]. 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 letter, we extend the histogram modification technique using pixel differences to increase hiding capacity. We use a binary tree structure to eliminate the requirement to communicate pairs of peak and zero points to the recipient. We also adopt a histogram shifting technique to prevent overflow and underflow.

To make this letter self-contained, Section II contains a detailed exposition of the proposed algorithm. In Section III, we experimentally investigate the relationship between the capacity and distortion, and the influence of variant images on the capacity. We also give compare performance with existing reversible schemes in the same section. Finally, we conclude the letter in Section IV.

II. Proposed Scheme

In [18], Ni *et al*. introduced a reversible data hiding scheme based on histogram modification using pairs of peak and zero points. 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. Once a pixel with value P is encountered, if the message bit is "1," the pixel value is increased by 1. Otherwise, no modification is needed. Data extraction is actually the reverse of the data hiding process. Note that the number of message bits that can be embedded into an image equals the number of pixels associated with the peak point.

However, the histogram modification technique does not work well when an image has an equal histogram. While multiple pairs of peak and minimum points can be used for embedding, the pure payload is still a little low. Moreover, the

Manuscript received May 14, 2008; revised August 25, 2008. First version published March 16, 2009; current version published June 19, 2009. This paper was recommended by Associate Editor R. C. Lancini.

W. -L. Tai is with the Department of Computer Science and Information, National Chung Cheng University, Chiayi 621, Taiwan (e-mail: taiwl@cs.ccu.edu.tw).

C. -M. Yeh and C. -C. Chang are with Department of Information Engineering and Computer Science, Feng Chia University, Taichung 407, Taiwan (e-mail: ccc@cs.ccu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org

Digital Object Identifier 10.1109/TCSVT.2009.2017409

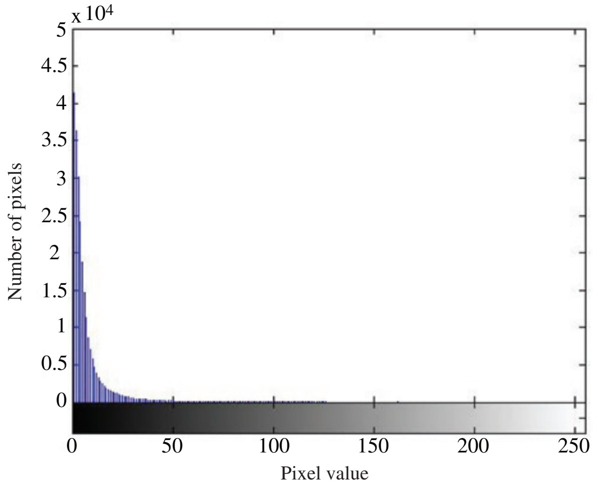

Fig. 1. Distribution of differences.

histogram modification technique carries with it an unsolved issue in that multiple pairs of peak and minimum points must be transmitted to the recipient via a side channel to ensure successful restoration.

Thus, we present an efficient extension of the histogram modification technique by considering the differences between adjacent pixels instead of simple pixel value. Since image neighbor pixels are strongly correlated, the distribution of pixel difference has a prominent maximum, that is, the difference is expected to be very close to zero, as shown in Fig. 1. We can find that the differences have almost a zero-mean and Laplacian-like distribution [21]. Distributions of other images also follow this model. Laplacian data can be applied to data hiding schemes [22]–[24] to improve their embedding ability. This observation leads us toward designs in which the embedding is done in pixel differences. We also use a tree structure to solve the issue of communicating multiple pairs of peak points to recipients. Having explained our background logic, we now outline the principle of the proposed reversible data hiding algorithm.

A. Histogram Modification on Pixel Differences

For an N-pixel 8-bit grayscale host image H with a pixel value x_i, where x_i denotes the grayscale value of the ith pixel, $0 \le i \le N-1, x_i \in \mathbf{Z}, x_i \in 0, 255$.

1) Scan the image H in an inverse s-order. Calculate the pixel difference d_i between pixels x_{i-1} and x_i by

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

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

3) Scan the whole image in the same inverse s-order as in Step 1. If $d_i > P$, shift x_i by 1 unit

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

where y_i is the watermarked value of pixel i.

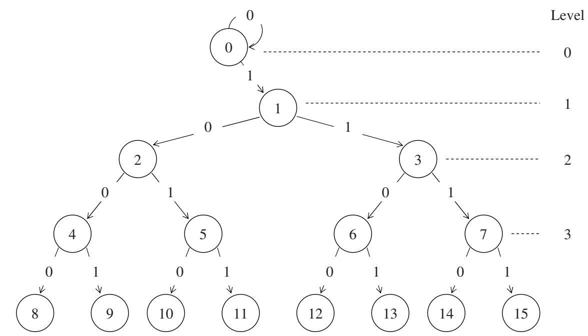

Fig. 2. Auxiliary binary tree for the proposed scheme.

4) if $d_i = P$, modify x_i according to the message bit

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

where b is a message bit to be embedded.

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 message bit b can be extracted by

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

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

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

Thus, an exact copy of the original host image is obtained. These steps complete the data hiding process in which only one peak point is used. Large hiding capacities can be obtained by repeating the data hiding process. However, recipients may not be able to retrieve both the embedded message and the original host image without knowledge of the peak points of every hiding pass. Thus, we present a binary tree structure in the following subsection that deals with communication of multiple peak points.

B. Binary Tree Structure

Fig. 2 shows an auxiliary binary tree for solving the issue of communication of multiple peak points. Each element denotes a peak point. Let us assume that the number of peak points used to embed messages is 2^L, where L is the level of the binary tree. Once a pixel difference d_i that satisfies $d_i < 2^L$ is encountered, if the message bit to be embedded is "0," the left child of the node d_i is visited; otherwise, the right child of the node d_i is visited. Higher payloads require the use of higher tree levels, thus quickly increasing the distortion in the image beyond acceptable levels. However, all the recipient needs to share with the sender is the tree level L, because we propose an auxiliary binary tree that predetermines multiple peak points used to embed messages. A detailed embedding algorithm with the auxiliary binary tree is given later in this letter.

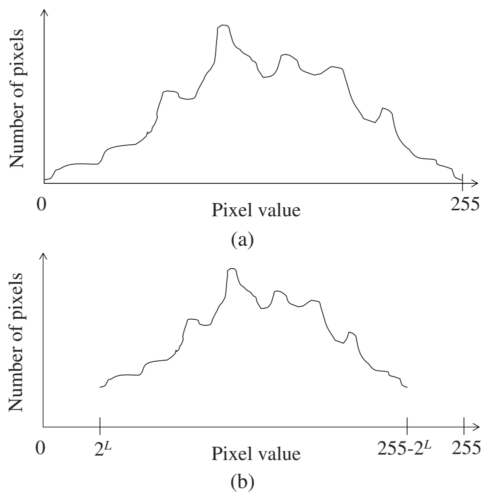

Fig. 3. Histogram shifting. (a) Original histogram. (b) Histogram shifting.

C. Prevent Overflow and Underflow

Modification of a pixel may not be allowed if the pixel is saturated (0 or 255). To prevent overflow and underflow, we adopt a histogram shifting technique that narrows the histogram from both sides, as shown in Fig. 3. Let us assume that the number of peak points used to embed messages is 2^L, where L is the level of the proposed binary tree structure. Thus, we shift the histogram from both sides by 2^L units to prevent overflow and underflow since the pixel x_i that satisfies $d_i \geq 2^L$ will shift by 2^L units after embedding takes place.

After narrowing the histogram to the range $2^L, 255 - 2^L$, we must record the histogram shifting information as overhead bookkeeping information. For this purpose, we create a one-bit map as the location map, which is equal in size to the host image. For a pixel having grayscale value in the range $2^L, 255 - 2^L$, 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 will yield a large increase in compression ability since pixels out of the range $2^L, 255 - 2^L$ are few and are almost always contiguous. The overhead information will be embedded into the host image together with the embedded message. Note that the maximum modification to a pixel is limited to 2^L according to the proposed tree structure. As a result, shifting the histogram from both sides by 2^L units enables us to avoid the occurrence of overflow and underflow.

D. Embedding Process

For an N-pixel 8-bit grayscale host image H with a pixel value x_i, where x_i denotes the grayscale value of the ith pixel, $0 \leq i \leq N-1$, $x_i \in \mathbf{Z}$, $x_i \in 0, 255$.

1) Determine the level L of the binary tree.
2) Shift the histogram from both sides by 2^L units. Note that the histogram shifting information is recorded as overhead bookkeeping information that will be embedded into the image itself with payload.
3) Scan the image H in an inverse s-order. Calculate the pixel difference d_i between pixels x_{i-1} and x_i.
4) Scan the whole image in the same inverse s-order. If $d_i \geq 2^L$, shift x_i by 2^L units
$$y_i = \begin{cases} x_i, & \text{if } i = 0 \\ x_i + 2^L, & \text{if } d_i \geq 2^L \text{ and } x_i \geq x_{i-1} \\ x_i - 2^L, & \text{if } d_i \geq 2^L \text{ and } x_i < x_{i-1} \end{cases}$$
where y_i is the watermarked value of pixel i.
5) If $d_i < 2^L$, modify x_i according to the message bit
$$y_i = \begin{cases} x_i + (d_i + b), & \text{if } x_i \geq x_{i-1} \\ x_i - (d_i + b), & \text{if } x_i < x_{i-1} \end{cases}$$
where b is a message bit to be embedded and b $\{0, 1\}$.

Note that the overhead information is included in the image itself with payload. Thus, the real capacity Cap that is referred to as pure payload is $Cap = N_p - |O|$, where N_p is the number of pixels that are associated with peak points and $|O|$ is the length of the overhead information.

E. Extraction Process

This process extracts both overhead information and payload from the watermarked image and losslessly recovers the host image. Let L be the level of the proposed binary tree. For an N-pixel 8-bit watermarked image W with a pixel value y_i, where y_i denotes the grayscale value of the ith pixel, $0 \leq i \leq N-1$, $y_i \in \mathbf{Z}$, $y_i \in 0, 255$.

1) Scan the watermarked image W in an inverse s-order.
2) If $|y_i x_{i-1}| < 2^{L+1}$, extract message bit b by
$$b = \begin{cases} 0, & \text{if } |y_i - x_{i-1}| \text{ is even} \\ 1, & \text{if } |y_i - x_{i-1}| \text{ is odd} \end{cases}$$
where x_{i-1} denotes the restored value of y_{i-1}.
3) Restore the original value of host pixel x_i by
$$x_i = \begin{cases} y_i + \left\lceil \dfrac{|y_i - x_{i-1}|}{2} \right\rceil \\ \quad \text{if } |y_i - x_{i-1}| < 2^{L+1} \text{ and } y_i < x_{i-1} \\ y_i - \left\lceil \dfrac{|y_i - x_{i-1}|}{2} \right\rceil \\ \quad \text{if } |y_i - x_{i-1}| < 2^{L+1} \text{ and } y_i > x_{i-1} \\ y_i + 2^L, \quad \text{if } |y_i - x_{i-1}| \geq 2^{L+1} \text{ and } y_i < x_{i-1} \\ y_i - 2^L, \quad \text{if } |y_i - x_{i-1}| \geq 2^{L+1} \text{ and } y_i > x_{i-1} \\ y_i, \quad \text{otherwise.} \end{cases}$$
4) Repeat Step 2 until the embedded message is completely extracted.
5) Extract the overhead information from the extracted message. If a value 1 is assigned in the location i, restore x_i to its original state by shifting it by 2^L units; otherwise, no shifting is required.

III. Experimental Results

To obtain a better understanding of how different host 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 commonly used grayscale images sized 512×512, "Lena," "Mandrill," "Boat," "Jet," "Pepper," and "GoldHill."

TABLE I
HIDING CAPACITY AND DISTORTION FOR TEST IMAGES WITH $L = 0$

Host image (512 × 512)	N_p	*Cap* (bits)	Overhead $\|O\|$ (bits)	PSNR (dB)	Bit rate (b/pixel)
Lena	22397	22377	20	48.32	0.0854
Mandrill	9938	9818	120	48.21	0.0375
Boat	25432	25412	20	48.35	0.0969
Jet	45492	45472	20	48.53	0.1734
Pepper	33413	33393	20	48.42	0.1273
GoldHill	18612	18592	20	48.29	0.0709

TABLE II
PURE PAYLOAD FOR TEST IMAGES WITH TREE LEVEL L

Host image (512 × 512)	Pure payload *Cap* for tree level $L = 0, 1, \ldots, 5$					
	0	1	2	3	4	5
Lena	22377	63117	129428	198557	237215	241399
Mandrill	9818	29130	63152	114108	173200	212131
Boat	25412	73422	143607	199877	230804	167167
Jet	45472	113107	182241	223420	244003	255025
Pepper	33393	90329	165621	199630	197532	198682
GoldHill	18592	54081	114709	185838	234786	215498
Average PSNR (dB)	48.35	42.83	37.98	33.94	30.47	26.62

A. Capacity versus Distortion Performance

Table I gives an example of how different images influence the pure payload *Cap* and the distortion at tree level $L = 0$. We can see a very high variability in N_p, which is the number of pixels 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 for test images there are no pixels outside the range [1, 254] except for the image "Baboon." Hence, the length of overhead information is 20 bits since the location map is losslessly compressed by the run-length coding algorithm. Further, the PSNR for the watermarked images in Table I closely matches our theoretical estimated lower bound.

Table II shows several experiments designed to see how the pure payload and distortion change with different tree levels. We note that images with abundant highly textured and noisy areas have generally smaller capacities. It is also very apparent that the pure payload increases very fast with different tree levels. For some images, however, the pure payload abruptly decreases with increased tree level since the length of overhead information increases very fast with the tree level.

The PSNR of the watermarked images is plotted against the pure payload size for test images in Fig. 4. The real capacity that can be achieved depends on the nature of the image itself. A nice feature of smooth images is that they provide higher real capacity at the same embedding distortion value. As a result, images with high correlation offer better performance than images with low correlation.

Fig. 5 shows the visual impacts of watermarked images at various real hiding capacities for "Lena" and "Mandrill." In general, the watermarked image hardly can be distinguished from the original image. For the smooth image "Lena" while the pure payload size is close to 1 b/pixel, the visual

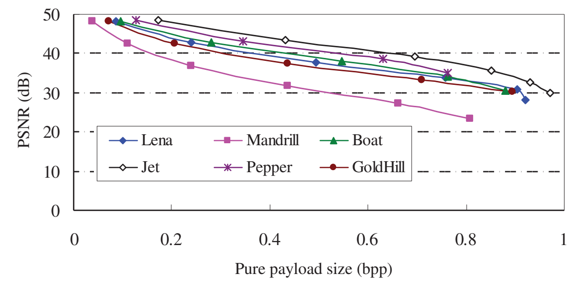

Fig. 4. PSNR versus pure payload size for test images.

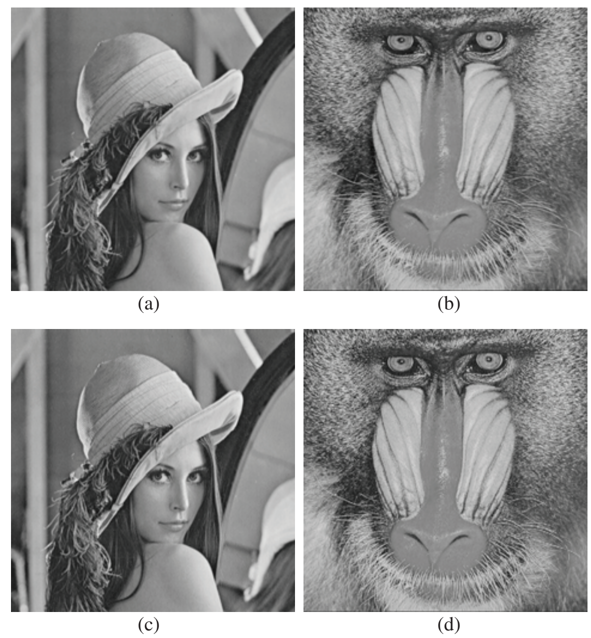

Fig. 5. Watermarked "Lena" and "Mandrill" images. (a) 48.32 dB embedded with 0.0854 b/pixel. (b) 48.21 dB embedded with 0.0375 b/pixel. (c) 30.75 dB embedded with 0.9049 b/pixel. (d) 23.49 dB embedded with 0.8092 b/pixel.

distortion is still quite small and the PSNR is higher than 30 dB. As a consequence, images with high textured areas and low correlation, such as "Mandrill," produce less N_p than do smooth images like "Lena," and thus embed less pure payload size at lower PSNR.

B. Comparison With Other Recent Schemes

Fig. 6 compares the pure payload of the "Lena" image in b/pixel versus image quality in PSNR delivered by the proposed scheme and other existing reversible schemes [7]–[9], [18]–[20]. Note that the proposed scheme and schemes [7], [18]–[20] are proposed in the spatial domain, whereas schemes [8], [9] are presented in the transform domain. Fig. 6 also shows the results of the proposed scheme with multiple layered hiding passes. Schemes [8], [9] achieved performance similar to our proposed scheme; however, their algorithms were performed in the wavelet domain.

Schemes [18]–[20] are presented based on histogram modification; nevertheless, their algorithms did not provide a solution to the problem of communicating multiple pairs of peak and minimum points. As Fig. 6 shows, Ni *et al.*'s scheme [18] has low hiding capacity compared to those of

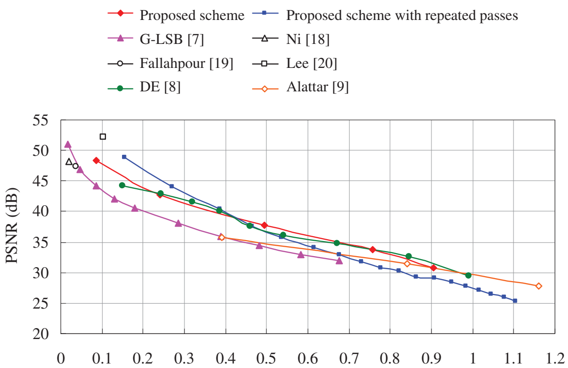

Fig. 6. Performance comparison for the "Lena" image with existing reversible schemes [7]–[9], [18]–[20].

the others. Fallahpour *et al*. [19] and Lee *et al*. [20] improved on Ni *et al*.'s work and derived better performance. However, they did not include the overhead information of histogram modification in the image itself with the payload. The evaluation results show that the proposed scheme achieves relatively high pure capacity with low distortion than existing schemes based on histogram modification.

IV. Conclusion

In this letter, we have presented an efficient extension of the histogram modification technique by considering the differences between adjacent pixels rather than simple pixel value. One common drawback of virtually all histogram modification techniques is that they must provide a side communication channel for pairs of peak and minimum points. To solve this problem, we introduced a binary tree that predetermines the multiple peak points used to embed messages; thus, the only information the sender and recipient must share is the tree level L. In addition, since neighbor pixels are often highly correlated and have spatial redundancy, the differences have a Laplacian-like distribution. This enables us to achieve large hiding capacity while keeping embedding distortion low.

References

[1] M. Wu and B. Lin, "Data hiding in image and video: Part I-fundamental issues and solutions," *IEEE Trans. Image Process.*, vol. 12, no. 6, pp. 685–695, Jun. 2003.

[2] M. Wu, H. Yu, and B. Liu, "Data hiding in image and video: Part II-designs and applications," *IEEE Trans. Image Process.*, vol. 12, no. 6, pp. 696–705, Jun. 2003.

[3] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," in *Proc. SPIE Security Watermarking Multimedia Contents III*, San Jose, CA, Jan. 2001, vol. 4314, pp. 197–208.

[4] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding for all image formats," in *Proc. SPIE Security Watermarking Multimedia Contents IV*, San Jose, CA, Jan. 2002, vol. 4675, pp. 572–583.

[5] C. De Vleeschouwer, J. F. Delaigle, and B. Macq, "Circular interpretation of bijective transformations in lossless watermarking for media asset management," *IEEE Trans. Multimedia*, vol. 5, no. 1, pp. 97–105, Mar. 2003.

[6] M. U. Celik, G. Sharma, and A. M. Tekalp, "Lossless watermarking for image authentication: A new framework and an implementation," *IEEE Trans. Image Process.*, vol. 15, no. 4, pp. 1042–1049, Apr. 2006.

[7] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 253–266, Feb. 2005.

[8] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.

[9] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Trans. Image Process.*, vol. 13, no. 8, pp. 1147–1156, Aug. 2004.

[10] L. Kamstra and H. J. A. M. Heijmans, "Reversible data embedding into images using wavelet techniques and sorting," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2082–2090, Dec. 2005.

[11] D. M. Thodi and J. J. Rodríguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 721–730, Mar. 2007.

[12] M. Van der Veen, F. Bruekers, A. Van Leest, and S. Cavin, "High capacity reversible watermarking for audio," in *Proc. SPIE Security Watermarking Multimedia Contents V*, Santa Clara, CA, Jan. 2003, vol. 5020, pp. 1–11.

[13] D. Rui and J. Fridrich, "Lossless authentication of MPEG-2 video," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2. Rochester, NY, 2002, pp. 893–896.

[14] J. Dittmann and O. Benedens, "Invertible authentication for 3-D meshes," in *Proc. SPIE Security Watermarking Multimedia Contents V*, Santa Clara, CA, Jan. 2003, vol. 5020, pp. 653–664.

[15] Y. Hu and B. Jeon, "Reversible visible watermarking and lossless recovery of original images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 11, pp. 1423–1429, Nov. 2006.

[16] C. C. Chang, W. L. Tai, and C. C. Lin, "A reversible data hiding scheme based on side match vector quantization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 10, pp. 1301–1308, Oct. 2006.

[17] S. Lee, C. D. Yoo, and T. Kalker, "Reversible image watermarking based on integer-to-integer wavelet transform," *IEEE Trans. Inf. Forensic Secur.*, vol. 2, no. 3, pp. 321–330, Sep. 2007.

[18] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar. 2006.

[19] M. Fallahpour and M. H. Sedaaghi, "High capacity lossless data hiding based on histogram modification," *IEICE Electron. Exp.*, vol. 4, no. 7, pp. 205–210, Apr. 2007.

[20] S. K. Lee, Y. H. Suh, and Y. S. Ho, "Reversible image authentication based on watermarking," in *Proc. IEEE Int. Conf. Multimedia Expo*, Toronto, ON, Canada, Jul. 2006, pp. 1321–1324.

[21] R. M. Norton, "The double exponential distribution: Using calculus to find a maximum likelihood estimator," *Amer. Stat.*, vol. 38, no. 2, pp. 135–136, May 1984.

[22] D. Zou, Y. Q. Shi, Z. Ni, and W. Su, "A semi-fragile lossless digital watermarking scheme based on integer wavelet transform," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 10, pp. 1294–1300, Oct. 2006.

[23] 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 Trans. Circuits Syst. Video Technol.*, vol. 18, no. 4, pp. 497–509, Apr. 2008.

[24] H. J. Kim, V. Sachnev, Y. Q. Shi, J. Nam, and H. G. Choo, "A novel difference expansion transform for reversible data embedding," *IEEE Trans. Inf. Forensic Secur.*, vol. 3, no. 3, pp. 456–465, Sep. 2008.