Reversible Watermarking Based on Invariant Relation of Three Pixels

Shaowei Weng[1], Shu-Chuan Chu[2], Jeng-Shyang Pan[3], and Lakhmi C. Jain[2]

[1] Information Engineering College, Guangdong University of Technology, P.R. China
wswweiwei@126.com
[2] School of Electrical and Information Engineering, University of South Australia
scchu@bit.kuas.edu.tw
[3] Harbin Institute of Technology Shenzhen Graduate School,P.R. China
jengshyangpan@gmail.com

Abstract. In Lin's method [1], images are divided into non-overlapping three-pixel blocks. Each pixel block contains two pairs composed of two neighboring pixels. Absolute differences between pairs are calculated. And meanwhile, the absolute difference having the highest occurrence rate is obtained. 1-bit watermark is embedded into a pixel pair whose absolute difference is equal to this obtained value. Since each pixel block contains two differences, Lin's method achieves embedding rate of at most $\frac{2}{3}$ bpp (bit per pixel) for a single embedding process. With the aim of further increasing embedding rate, we modify the embedding process in Lin's method to keep the third pixel of a pixel block unaltered in the proposed method. Then, this unchanged pixel is used again to reform into a new pixel block with its two neighboring pixels. As a result, the embedding rate can reach to 1 bpp for a single embedding process. ...

1 Introduction

For some critical applications such as the fields of the law enforcement, medical and military image system, it is crucial to restore the original image without any distortions. The watermarking techniques satisfying these requirements are referred to as the reversible watermarking. The reversible watermarking is designed so that it can be removed to completely restore the original image.

The concept of reversible watermark firstly appeared in the patent owned by Eastman Kodak [2]. Several researchers had developed reversible watermarking [1, 3–10]. Tian [8] expanded the differences between two neighboring pixel values to embed a bit into each pixel pair without causing overflows/underflows after expansion. Alatter [9] used the difference expansion of quads to embed a large amount of data into the grayscale images. His algorithm allowed three bits to be embedded into every quad. Yang *et al.* [10] proposed a simple lossless data

J.-S. Pan, S.-M. Chen, and N.T. Nguyen (Eds.): ICCCI 2010, Part III, LNAI 6423, pp. 71–80, 2010.

hiding method based on the coefficient-bias algorithm by embedding bits in both spatial domain and frequency domain.

In Lin's method [1], images are divided into non-overlapping three-pixel blocks. Each pixel block contains two pairs composed of two neighboring pixels. Absolute differences between pairs are calculated. And meanwhile, the absolute difference having the highest occurrence rate is obtained. 1-bit watermark is embedded into a pixel pair whose absolute difference is equal to this obtained value. Since each pixel block contains two differences, Lin's method achieves embedding rate of at most $\frac{2}{3}$ bpp (bit per pixel) for a single embedding process. With the aim of further increasing embedding rate, the embedding process in Lin's method is changed to keep the third pixel of a pixel block unaltered in the proposed method. Then, this unchanged pixel is used again to form into a new pixel block with its two neighboring pixels. As a result, the embedding rate can reach to 1 bpp for a single embedding process.

The remains of the paper are organized as follows. In Section 2, the proposed method is introduced. Watermark embedding and data extracting and image restoration are presented in Section 2.1 and 2.3, respectively. The performance analysis is discussed in Section 2.2. The experimental results are shown in Section 3 and finally we conclude the paper in Section 4.

2 The Proposed Method

Let $p = (p_1, p_2, p_3)$ denote a pixel block containing three neighboring pixels. Each p contains two absolute differences d_1, d_2, i.e., $d_1 = |p_1 - p_2|$, $d_2 = |p_2 - p_3|$. For a $\mathtt{W} \times \mathtt{H}$-size image, it is partitioned into non-overlapping three-pixel blocks in Lin's method, thus the number of differences occupy only $\frac{2}{3}$ of image size. In the proposed method, with the aim of further increasing the number of differences, p_3 for any three-pixel block p will be kept unchanged in the embedding process, in such case, p_3 can be used again and be reformed into a new three-pixel block with its two right neighboring pixels. Since p_3 is used twice, the number of differences approaches to image size.

Let $g(d)$ denote the number of differences whose absolute values equal d, where $0 \le d \le 253$. M_d and m_d are respectively used to represent the absolute values of differences having the largest and the small occurrence, i.e., $g(M_d) \ge g(M^{'})$ and $g(m_d) \le g(m^{'})$ for $0 \le M^{'}, m^{'} \le 253$. In Lin's method, 1-bit watermark is embedded into difference whose absolute value is equal to M_d. For a pixel block p, if $d_1 = M_d$ and $d_2 = M_d$, then this p is capable of carrying two-bit watermark information. In the proposed method, in order to increase the number of differences used for embedding, we modify Lin's method to keep p_3 unaltered. For p, there exists five relations (i.e., $p_1 > p_2 > p_3$, $p_1 < p_2 < p_3$, $p_1 = p_2 = p_3$, $p_1 < p_2 > p_3$ and $p_1 > p_2 < p_3$) among p_1, p_2 and p_3. Take $p_1 > p_2 > p_3$ for example, since $d_1 = M_d$ and $d_2 = M_d$, after watermark embedding, $p_1^{'} = p_1 + w_1 + w_2$, $p_2^{'} = p_2 + w_2$ and $p_3^{'} = p_3$. Note that no change to relation among $p_1^{'}$, $p_2^{'}$ and $p_3^{'}$ occurs before and after watermark embedding, i.e., $p_1^{'} > p_2^{'} > p_3^{'}$. $d_1^{'}$ and $d_2^{'}$ are respectively used to denote the watermarked differences of d_1

and d_2. After watermark embedding, $d_1^{'} = p_1^{'} - p_2^{'} = p_1 - p_2 + w_1 = d_1 + w_1$, $d_2^{'} = p_2^{'} - p_3^{'} = p_2 - p_3 + w_2 = d_2 + w_2$. Hence, in the extraction process, by comparing $d_1^{'}$ (or $d_2^{'}$) with M_d and m_d, 2-bit watermark can be correctly extracted. Detailed modification is illustrated in Procedure 1.

2.1 Watermark Embedding

In the watermark embedding process, 1-bit watermark is embedded into differences whose absolute value equals M_d. Since each p have two absolute differences d_1 and d_2. d_1 is divided into three intervals (i.e., $d_1 = M_d$, $m_d \geq d_1 > M_d$ and $d_1 < M_d$ `or` $d_1 > m_d$) according to its value. Similarly, d_2 is also divided into three intervals (i.e., $d_2 = M_d$, $m_d \geq d_2 > M_d$ and $d_2 < M_d$ `or` $d_2 > m_d$). d_1 and d_2 are combined to generate nine combinations. Nine combinations respectively correspond to nine different embedding strategy detailedly shown in the following embedding procedure.

```
if d1 == Md
{
    if d2 == Md
        call embed_2_bits;
    else
        {
        if Md < d2 <= md
            call embed_1_bit_and_increase_difference;
        else
            call embed_1_bit_and_leave_unchanged;
        }
    }
elseif Md < d1 <= md
{
    if d2 == Md
        call increase_difference_and_embed_1_bit;
    else
    {
        if Md < d2 < md
            call increase_2_difference;
        else
            increase_difference_and_leave_unchanged;
        }
    }
else
{
    if d2 == Md
        call leave_unchanged_and_embed_1_bit;
    else
```

```
    {
    if M_d < d_2 < m_d
        call leave_unchanged_and_increase_difference;
    else
        Do nothing;
    }
}
```

When $m_d < d_1$ `or` $d_1 < M_d$ and $m_d < d_2$ `or` $d_2 < M_d$, we do nothing. Each of the remaining eight strategy will call a function to realize the embedding process. Eight functions are respectively illustrated in Procedure 1 to Procedure 8. With the help of Table 1 to Table 8, p can be correctly retrieved in the extraction process by comparing $d_1^{'}$ (or $d_2^{'}$) with M_d and m_d.

Procedure 1. Embedding_two_bits

$p_1 > p_2 > p_3$	$p_1^{'} = p_1 + w_1 + w_2, p_2^{'} = p_2 + w_2$
$p_1 < p_2 < p_3$	$p_1^{'} = p_1 - w_1 - w_2, p_2^{'} = p_2 - w_2$
$p_1 = p_2 = p_3$	if $w_1 == 1$ and $w_2 == 1$ $p_2^{'} = p_2 + 1$ elseif $w_1 == 0$ and $w_2 == 1$ $p_1^{'} = p_1 - 1, p_2^{'} = p_2 - 1$ else $p_1^{'} = p_1 - w_1, p_2^{'} = p_2 - w_2$
$p_1 < p_2 > p_3$	if $w_1 == 1$ and $w_2 == 1$ $p_2^{'} = p_2 + 1$ elseif $w_1 == 0$ and $w_2 == 1$ $p_1^{'} = p_1 + 1, p_2^{'} = p_2 + 1$ else $p_1^{'} = p_1 - w_1, p_2^{'} = p_2 + w_2$
$p_1 > p_2 < p_3$	if $w_1 == 1$ and $w_2 == 1$ $p_2^{'} = p_2 - 1$ elseif $w_1 == 0$ and $w_2 == 1$ $p_1^{'} = p_1 - 1, p_2^{'} = p_2 - 1$ else $p_1^{'} = p_1 + w_1, p_2^{'} = p_2 - w_2$

Table 1. Changes to Procedure 1 after watermark embedding

<table>
<tr><td>$p_1 > p_2 > p_3$</td><td>$p_1^{'} > p_2^{'} > p_3^{'}$</td><td rowspan="5">$d_1^{'} \in \{M_d, M_d + 1\}$</td><td rowspan="5">$d_2^{'} \in \{M_d, M_d + 1\}$</td></tr>
<tr><td>$p_1 < p_2 < p_3$</td><td>$p_1^{'} > p_2^{'} > p_3^{'}$</td></tr>
<tr><td>$p_1 = p_2 = p_3$</td><td>$p_1 \neq p_2^{'} \neq p_3^{'}$</td></tr>
<tr><td>$p_1 < p_2 > p_3$</td><td>$p_1^{'} < p_2^{'} > p_3^{'}$</td></tr>
<tr><td>$p_1 > p_2 < p_3$</td><td>$p_1^{'} > p_2^{'} < p_3^{'}$</td></tr>
</table>

Procedure 2. Embed_1_bit_and_increase_difference

$p_1 > p_2 > p_3$	$p_1^{'} = p_1 + w_1 + 1, p_2^{'} = p_2 + 1$
$p_1 < p_2 < p_3$	$p_1^{'} = p_1 - w_1 - 1, p_2^{'} = p_2 - 1$
$p_1 < p_2 > p_3$	if $w_1 == 1$ $p_2^{'} = p_2 + 1$ else $p_1^{'} = p_1 + 1, p_2^{'} = p_2 + 1$
$p_1 > p_2 < p_3$	if $w_1 == 1$ $p_2^{'} = p_2 - 1$ else $p_1^{'} = p_1 - 1, p_2^{'} = p_2 - 1$

Table 2. Changes to Procedure 2 after watermark embedding

<table>
<tr><td>$p_1 > p_2 > p_3$</td><td>$p_1^{'} \geq p_2^{'} > p_3^{'}$</td><td rowspan="4">$d_1^{'} \in \{M_d, M_d + 1\}$</td><td rowspan="4">$M_d + 1 < d_2^{'} < m_d + 1$</td></tr>
<tr><td>$p_1 < p_2 < p_3$</td><td>$p_1^{'} \leq p_2^{'} > p_3^{'}$</td></tr>
<tr><td>$p_1 < p_2 > p_3$</td><td>$p_1^{'} < p_2^{'} > p_3^{'}$</td></tr>
<tr><td>$p_1 > p_2 < p_3$</td><td>$p_1^{'} > p_2^{'} < p_3^{'}$</td></tr>
</table>

Procedure 3. Embed_1_bit_and_leave_unchanged

$p_1 > p_2$	$p_1^{'} = p_1 + w_1$
$p_1 \leq p_2$	$p_1^{'} = p_1 - w_1$

Table 3. Changes to Procedure 3 after watermark embedding

<table>
<tr><td>$p_1 > p_2$</td><td>$p_1^{'} > p_2^{'}$</td><td rowspan="2">$d_1^{'} \in \{M_d, M_d + 1\}$</td><td rowspan="2">$d_2^{'} < M_d$ or $d_2^{'} > m_d$</td></tr>
<tr><td>$p_1 < p_2$</td><td>$p_1^{'} < p_2^{'}$</td></tr>
</table>

Procedure 4. Increase_difference_and_embed_1_bit

$p_1 > p_2 > p_3$	$p_1^{'} = p_1 + 1 + w_2, p_2^{'} = p_2 + w_2$
$p_1 < p_2 < p_3$	$p_1^{'} = p_1 - 1 - w_2, p_2^{'} = p_2 - w_2$
$p_1 < p_2 > p_3$	if $w_2 == 1$ $p_2^{'} = p_2 + 1$ else $p_1^{'} = p_1 - 1$
$p_1 > p_2 < p_3$	if $w_2 == 1$ $p_2^{'} = p_2 - 1$ else $p_1^{'} = p_1 + 1,$

Table 4. Changes to Procedure 4 after watermark embedding

$p_1 > p_2 > p_3$	$p_1^{'} > p_2^{'} > p_3^{'}$	$M_d + 1 < d_1^{'} < m_d + 1$	$d_2^{'} \in \{M_d, M_d + 1\}$
$p_1 < p_2 < p_3$	$p_1^{'} > p_2^{'} > p_3^{'}$		
$p_1 < p_2 > p_3$	$p_1^{'} < p_2^{'} > p_3^{'}$		
$p_1 > p_2 < p_3$	$p_1^{'} > p_2^{'} < p_3^{'}$		

Procedure 5. Increase_2_differences

$p_1 > p_2 > p_3$	$p_1^{'} = p_1 + 2, p_2^{'} = p_2 + 1$
$p_1 < p_2 < p_3$	$p_1^{'} = p_1 - 2, p_2^{'} = p_2 - 1$
$p_1 < p_2 > p_3$	$p_2^{'} = p_2 + 1$
$p_1 > p_2 < p_3$	$p_2^{'} = p_2 - 1$

Table 5. Changes to Procedure 5 after watermark embedding

$p_1 > p_2 > p_3$	$p_1^{'} > p_2^{'} > p_3^{'}$	$M_d + 1 < d_1^{'} < m_d + 1$	$M_d + 1 < d_2^{'} < m_d + 1$
$p_1 < p_2 < p_3$	$p_1^{'} > p_2^{'} > p_3^{'}$		
$p_1 < p_2 > p_3$	$p_1^{'} < p_2^{'} > p_3^{'}$		
$p_1 > p_2 < p_3$	$p_1^{'} > p_2^{'} < p_3^{'}$		

Procedure 6. increase_difference_and_leave_unchanged

$p_1 > p_2$	$p_1^{'} = p_1 + 1$
$p_1 \leq p_2$	$p_1^{'} = p_1 - 1$

Table 6. Changes to Procedure 6 after watermark embedding

$p_1 > p_2$	$p_1^{'} > p_2^{'}$	$M_d + 1 < d_1^{'} < m_d + 1$	$d_2^{'} < M_d$ or $d_2^{'} > m_d$
$p_1 < p_2$	$p_1^{'} < p_2^{'}$		

Procedure 7. Leave_unchanged_and_embed_1_bit

$p_2 > p_3$	$p_1^{'} = p_1 + w_2, p_2^{'} = p_2 + w_2$
$p_2 \leq p_3$	$p_1^{'} = p_1 - w_2, p_2^{'} = p_2 - w_2$

Table 7. Changes to Procedure 7 after watermark embedding

$p_2 > p_3$	$p_2^{'} > p_3^{'}$	$d_1^{'} < M_d$ or $d_1^{'} > m_d$	$d_2^{'} \in \{M_d, M_d + 1\}$
$p_2 < p_3$	$p_2^{'} < p_3^{'}$		

Procedure 8. Leave_unchanged_and_increase_difference

$p_2 > p_3$	$p_1^{'} = p_1 + 1, p_2^{'} = p_2 + 1$
$p_2 < p_3$	$p_1^{'} = p_1 - 1, p_2^{'} = p_2 - 1$

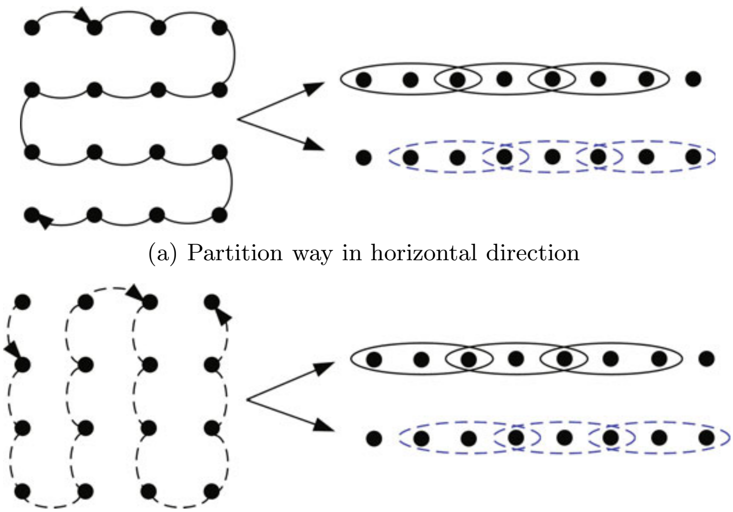
Table 8. Changes to Procedure 8 after watermark embedding

$p_2 > p_3$	$p_2^{'} > p_3^{'}$	$d_1^{'} < M_d$ or $d_1^{'} > m_d$	$M_d + 1 < d_2^{'} < m_d + 1$
$p_2 < p_3$	$p_2^{'} < p_3^{'}$		

I is converted into a one-dimension pixel list respectively according to the arrow directions shown in Fig. 1(a) and Fig. 1(b). Each list has two different ways (respectively marked by black and blue) to partition all pixels into overlapped three pixel blocks. For each way, we find the number of absolute differences having the maximum occurrence. Hence, two lists will generate four absolute differences with the maximum occurrence, respectively use M_{d1}, M_{d2}, M_{d3} and M_{d4} to denote them. Find the maximum value from M_{di}, $1 \leq i \leq 4$, and $i \in \mathbb{Z}$ and use M_d to denote this value. I is converted into a one-dimension pixel list I_{D1} according to the selected order.

Three consecutive pixels p_1, p_2 and p_3 of I_{D1} is grouped into a pixel block (p_1, p_2, p_3), where $0 \leq p_1 \leq 255$, $0 \leq p_2 \leq 255$ and $0 \leq p_3 \leq 255$. For each p, calculate its two differences d_1 and d_2, and then select the corresponding procedure from Procedure 1 to Procedure 8 according to the combination of d_1 and d_2, finally obtain $p_1^{'}$, $p_2^{'}$ and $p_3^{'}$ based on the relation among p_1,p_2 and p_3.

A location map L_M is generated and denoted by a bit sequence of size $W \times H$. For a pixel block p, if $0 \leq p_1^{'} \leq 255$, $0 \leq p_2^{'} \leq 255$ and $0 \leq p_3^{'} \leq 255$ after performing embedding procedure, this p is marked by '1' in the map L_M. otherwise by '0'. The location map is compressed losslessly by an arithmetic



(a) Partition way in horizontal direction

(b) Partition way in vertical direction

Fig. 1. Partition way for a 4 × 4 images block

encoder and the resulting bitstream is denoted by $\mathcal{L}$. L_S is the bit length of $\mathcal{L}$. Embedding procedure is performed if p is marked by '1' in L_M. Otherwise, p is kept unchanged.

After the first L_S pixels have been processed, their LSBs are replaced by $\mathcal{L}$, and then the L_S LSBs to be replaced are appended to $\mathcal{P}$. Finally, the rest of $\mathcal{P}$ and the L_S appended bits are embedded into the remaining unprocessed blocks. After all the blocks are processed, a new watermarked image I_W is obtained.

2.2 Performance Analysis

The embedding distortion caused by the embedding procedure is illustrated in this subsection. Following that is the definition of data-hiding capacity.

Modifications to Pixels. Take p with $d_1 = M_d$ and $d_2 = M_d$ for example. Since $d_1 = M_d$ and $d_2 = M_d$, then two-bit watermark information is embedded into p. If $p_1 > p_2 > p_3$, after watermark embedding, $p_1^{'} = p_1 + w_1 + w_2$, $p_2^{'} = p_2 + w_2$ and $p_3^{'} = p_3$. For pixel p_1, the difference d_{p1} between $p_1^{'}$ and p_1 equals $p_1^{'} - p_1 = w_1 + w_2$. Similarly, $d_{p2} = p_2^{'} - p_2 = w_2$, $d_{p3} = 0$. However, in Lin's method, for $p_1 > p_2 > p_3$, $d_{p1} = w_1$, $d_{p2} = 0$ and $d_{p3} = w_2$. Through the comparison of the proposed method and Lin's method, the modification to p_1 caused by watermark embedding is increased. To conclude from the above example, we increase the number of differences at the cost of enhancing the embedding distortions.

Hiding Capacity. The maximum hiding capacity D is given by:

$$D = g(M_d) - L_S \tag{1}$$

From Eq. (1), a part of the available capacity is consumed by $\mathcal{L}$.

2.3 Data Extraction and Image Restoration

The watermarked image I_W is converted into a one-dimensional pixel list in the same way as was done in embedding.

For I_W, LSBs of the first L_S watermarked pixels are collected into a bitstream $\mathcal{B}$. $\mathcal{B}$ are decompressed by an arithmetic decoder to retrieve the location map. By identifying the EOS symbol in $\mathcal{B}$, the bits from the start until EOS are decompressed by an arithmetic decoder to retrieve the location map. Data extraction and pixel restoration is carried out in inverse order as in embedding. M_d and m_d are transmitted to the receiving side. With help of M_d and m_d, the watermark sequence can be correctly extracted. For each pair $p^{'} = (p_1^{'}, p_2^{'}, p_3^{'})$, if $p^{'}$'s location is associated with '0' in the location map, then it is ignored. Otherwise, p can be retrieved in virtue of Table 1 to Table 8.

3 Experimental Results

The proposed method is implemented and tested on various standard test images using MATLAB. The performance for four most frequently used 512×512 grayscale standard test image is presented in Table 9, where we use number of bits and PSNR values for measurement for capacity and distortion.

We modify Lin's method by keeping p_3 of any pixel block p. As a result, the proposed method increases the number of differences whose absolute value equals M_d at the cost of enhancing the modifications to p_1 and p_2. Take Procedure 1 for example, for the situation of $p_1 > p_2 > p_3$, when $w_1 = 1$ and $w_2 = 1$, $d_1^{'} = 2$, $d_1^{'} = 1$. Therefore, decreasing the modifications to p_1 is the key to increasing the payload while reducing the distortions. We will be devoted to research in increasing the performance of the proposed method in the future work.

Table 9. Performance of the proposed method

Image	Embedding capacity (bits)	PSNR value (dB)
Baboon	19265	46.7062
Boat	50407	47.1568
Lena	54069	47.6528
Barbara	37454	46.7380

4 Conclusions

In Lin's method, images are divided into non-overlapping three-pixel blocks. Each pixel block contains two pairs composed of two neighboring pixels. Absolute differences between pairs are calculated. And meanwhile, the absolute difference having the highest occurrence rate is obtained. 1-bit watermark is embedded into a pixel pair whose absolute difference is equal to this obtained value. Since each pixel block contains two differences, Lin's method achieves embedding rate of at most $\frac{2}{3}$ bpp (bit per pixel) for a single embedding process. With the aim of further increasing embedding rate, we modify the embedding process in Lin's method to keep the third pixel of a pixel block unaltered in the proposed method. Then, this unchanged pixel is used again to reform into a new pixel block with its two neighboring pixels. As a result, the embedding rate can reach to 1 bpp for a single embedding process.

References

1. Lin, C.-C., Hsueh, N.-L.: A lossless data hiding scheme based on three-pixel block differences. Pattern Recognition 41(4), 1415–1425 (2008)
2. Honsinger, C.W., Jones, P., Rabbani, M., Stoffel, J.C.: Lossless recovery of an original image containing embedded data. US patent, No 77102/E-D (1999)
3. Goljan, M., Fridrich, J., Du, R.: Distortion-free data embedding for images. In: Moskowitz, I.S. (ed.) IH 2001. LNCS, vol. 2137, pp. 27–41. Springer, Heidelberg (2001)
4. Celik, M.U., Sharma, G., Tekalp, A.M., Saber, E.: Reversible data hiding. In: Proceedings of IEEE 2002 International Conference on Image Processing, vol. 2, pp. 157–160 (2002)
5. Xuan, G.R., Yang, C.Y., Zhen, Y.Z., Shi, Y.Q.: Reversible Data Hiding Using Integer Wavelet Transform and Companding Technique. In: Cox, I., Kalker, T., Lee, H.-K. (eds.) IWDW 2004. LNCS, vol. 3304, pp. 115–124. Springer, Heidelberg (2004)
6. Celik, M.U., Sharma, G., Tekalp, A.M.: Lossless watermarking for image authentication a new framework and an implementation. IEEE Trans. on Image Processing, 1042–1049 (2006)
7. Thodi, M., Rodrguez, J.J.: Prediction-error based reversible watermarking. In: Proc. of ICIP, Genova, vol. 3, pp. 1549–1552 (October 2004)
8. Tian, J.: Reversible data embedding using a difference expansion. IEEE Trans. Circuits and Systems for Video Tech. 13(8), 890–896 (2003)
9. Alattar, A.M.: Reversible watermark using difference expansion of quads. In: Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP 2004), vol. 3, pp. 377–380 (2004)
10. Yang, C.-Y., Hu, W.-C., Lin, C.-H.: Reversible data hiding by coefficient-bias algorithm. Journal of Information Hiding and Multimedia Signal Processing 1(2), 91–100 (2010)