World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:8, No:3, 2014

High Capacity Reversible Watermarking through Interpolated Error Shifting

Hae-Yeoun Lee

***Abstract*—**Reversible watermarking that not only protects the copyright but also preserve the original quality of the digital content have been intensively studied. In particular, the demand for reversible watermarking has increased. In this paper, we propose a reversible watermarking scheme based on interpolation-error shifting and error pre-compensation. The intensity of a pixel is interpolated from the intensities of neighboring pixels, and the difference histogram between the interpolated and the original intensities is obtained and modified to embed the watermark message. By restoring the difference histogram, the embedded watermark is extracted and the original image is recovered by compensating for the interpolation error. The overflow and underflow are prevented by error pre-compensation. To show the performance of the method, the proposed algorithm is compared with other methods using various test images.

***Keywords*—**Reversible watermarking, High capacity, High quality, Interpolated error shifting, Error pre-compensation.

I. Introduction

WITH the rapid spread of digital content, copyright protection has become critical. Digital watermarking is a solution for this purpose that embeds secret information into the host media to provide copyright protection [1], [2]. Robust watermarking techniques have been intensively studied to develop watermarks that can withstand common attacks while preserving the quality of the content. Since these techniques do not facilitate recovery of the original content after the watermark has been extracted, they are not useful for quality-sensitive applications such as medical, military, and artistic imaging.

To satisfy the requirements of quality-sensitive applications, reversible watermarking has been proposed as a solution [3], [4]. The main purpose of reversible watermarking is to facilitate exact recovery of the original image from the watermarked image after watermark removal and to simultaneously achieve a high capacity [5].

In this paper, we propose a reversible watermarking technique that is based on interpolation-error shifting and error pre-compensation. The intensity of a pixel is interpolated from the intensities of neighboring pixels, and the difference histogram between the interpolated and original intensities is calculated and modified to embed the watermark.

The peak of the maximum bin of the difference histogram increases when the interpolated pixel instead of the adjacent pixel is used for obtaining the histogram. Therefore, the high peak of the maximum bin ensures a high capacity. Also, error pre-compensation prevents overflow and underflow. By restoring the difference histogram, the embedded watermark is extracted and the original image is recovered by compensating for the interpolation error. In the experiment, the presented algorithm is compared with other reversible algorithms by applying the algorithms to various test images.

The paper is organized as follows. In Section II, previous reversible watermarking is summarized. In Section III, the proposed technique is presented. Experimental results are given in Section IV. Conclusions are presented in Section V.

Hae-Yeoun Lee is with the Kumoh National Institute of Technology, Gumi, Gyeongbuk, Republic of Korea (phone: 82-54-478-7548; fax: 82-54-478-7539; e-mail: haeyeoun.lee@kumoh.ac.kr).

II. Related Works

Reversible watermarking approaches can be categorized into four types, depending on the embedding strategy or the embedding domain.

A. Lossless Compression-Based Approaches

The methods in this category losslessly compress selected image features for acquiring extra space for watermark embedding. Fridrich et al. used a JBIG lossless compression scheme for compressing a proper bit plane and then they embedded an image hash by appending it to the compressed bit stream [6]. Celik et al. used a CALIC lossless compression algorithm and achieved a high capacity by using the generalized least significant bit embedding technique [7].

B. Transform Domain-Based Approaches

The main aim in transform domains is to ensure high visual quality. A modification in the transform domain does not cause a perceptible difference to the naked eye because a coefficient in the transform domain involves several pixels in the spatial domain and the modified energy is dispersed to several pixels. Studies have been performed in the transform domains such as discrete cosine transform (DCT) or discrete wavelet transform (DWT) domain, in which message bits are embedded into the corresponding coefficients. Yang et al. proposed a reversible watermarking algorithm that utilized a block-based integer DCT domain [8]. Lee et al. applied an integer-to-integer wavelet transform to image blocks and embedded message bits into the high-frequency wavelet coefficients of each block [9].

C. Difference Expansion-Based Approaches

Difference expansion (DE)-based algorithms are based on a local similarity between pixels in natural images. They utilize the differences between pixel pairs and embed watermarks by expanding the differences. The representative DE method was proposed by Tian [10] where an integer Haar wavelet transform was used to obtain high-pass components that were considered as the differences between pixel pairs. Message bits were embedded by expanding these differences. Alattar introduced a generalized DE-based technique that could be applied to any integer transform [11]. Kamstra and Heijmans extended the DE technique to predict which pixel will be expandable by employing low-pass components [12]. Coltuc designed a low distortion reversible watermarking based on prediction-error expansion [13]. Hu et al. proposed a DE-embedding algorithm that was based on an integer Haar wavelet transform [14].

D. Histogram Modification-Based Approaches

In histogram-modification techniques, an image histogram is used to obtain space and hide message bits by modifying the histogram. Ni et al. utilized the minimum and maximum bins of a given histogram [15]. Lin et al. divided an image into non-overlapping blocks and generated a difference image between each block [16]. Tsai et al. used a residue image that indicated the difference between the basic pixel and remainder pixels in a non-overlapping block [17]. Kim et al. presented a reversible watermarking algorithm that modified the difference histogram for sub-sampled images [3].

III. Proposed Reversible Watermarking Method

This section presents a histogram modification-based reversible watermarking for realizing a high capacity, high quality, and low computational complexity. Since it generates errors of only +1 or -1 when the coefficients of the histogram are shifted, imperceptibility is guaranteed. For achieving a high capacity, it is important to maximize the number of coefficients at the peak point in the histogram. In this paper, we utilize the interpolation error for achieving a higher capacity. Overflow or underflow is prevented by error pre-compensation.

A. Interpolation-Error Estimation

Histogram modification techniques calculate a histogram and select a maximum bin for making embedding spaces. Since watermarking capacity depends on the peak of the maximum bin, it is essential to maximize the highest bin.

For achieving a higher capacity and visual quality, the interpolated image is considered as the input for obtaining the difference histogram. The interpolation technique results in a high correlation between neighboring pixels and help to get the maximum bin in the histogram. It is important to maintain the same estimated image during both the embedding and detection processes. Therefore, it needs to select some pixels that are to be unchanged and fixed.

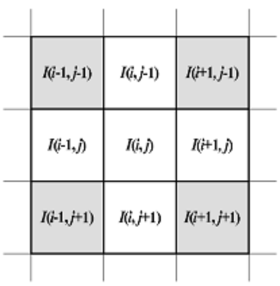

Fig. 1 Fixed pixels and interpolated pixels

As shown in Fig. 1, when we fix *I*(*i-1, j-1*), *I*(*i+1, j-1*), *I*(*i-1, j+1*), and *I*(*i+1, j+1*), *I*(*i, j-1*), *I*(*i, j*), and *I*(*i-1, j*) are interpolated as follows.

$$
\begin{aligned}
I_p(i, j-1) &= \{I(i-1, j-1) + I(i+1, j-1)\}/2, \\
I_p(i, j) &= \{I(i-1, j-1) + I(i+1, j-1) + I(i-1, j+1) + I(i+1, j+1)\}/4, \\
I_p(i-1, j) &= \{I(i-1, j-1) + I(i-1, j+1)\}/2.
\end{aligned}
$$

The interpolation error for *I*(*i, j*), i.e., the difference between the intensities of the original pixel and interpolated pixel, is calculated as

$$\delta(i, j) = \left| I(i, j) - I_p(i, j) \right|$$

The histogram for the interpolation error δ is then calculated by excluding the fixed pixels. Since the proposed difference histogram has a greater maximum frequency than the histogram of previous schemes, a high capacity can be achieved by using the proposed difference histogram.

B. Error Pre-Compensation

When a watermark is embedded into the difference histogram by modifying it, overflow or underflow can occur in the pixel domain. They cause salt-and-pepper noise, degrade the visual quality of the image, and destroy the reversibility of the proposed algorithm.

To preventing overflow and underflow, we use a location map in [10]. The pixels for which overflow and underflow can occur or has already occurred are marked on the location map. The marked pixels can be excluded in the embedding step or can be compensated for in the recovery step.

Repetitive embedding is required to achieve a high capacity. The pixels for which overflow and underflow can occur and their values vary with the repeat count. If the watermark embedding is repeated γ times, the pixels whose values exceed (255-γ) have a high probability of being overflowed and can be compensated by subtracting γ. Similarly, the pixels whose values are less than γ have a high probability of being underflowed and can be prevented by adding γ. In the detection process, the original image is restored by adding the compensated value γ to the value of the pixels marked on the location map. To reduce the capacity of the location map, JBIG1, which has a high compression efficiency, is adopted as the lossless image-compression standard [18].

C. Embedding Algorithm

Fig. 2 shows the entire watermark-embedding procedure that consists of several steps. The watermark is embedded by modifying the difference histogram for the compensated and interpolated images. The watermark to be embedded is assumed to be a pseudo-random binary sequence, and a grayscale image is employed as the input.

To prevent overflow and underflow, a pre-processing operation is performed on the input image *I*. The repeat count γ is used to compensate for the overflow and underflow as below.

$$I_c(i,j)=\begin{cases} I(i,j)-\gamma, & \text{if } I(i,j)>255-\gamma \\ I(i,j)+\gamma, & \text{if } I(i,j)<\gamma \\ I(i,j), & \text{otherwise} \end{cases}$$

M and N denote the height and width of the input image, respectively. The location of the compensated pixel (i, j) is marked on the location map μ.

$$\mu(i,j)=\begin{cases} 1, & \text{if } I(i,j)>255-\gamma \\ 1, & \text{if } I(i,j)<\gamma \\ 0, & \text{otherwise} \end{cases}$$

The compensated image I_c is used instead of the original image I.

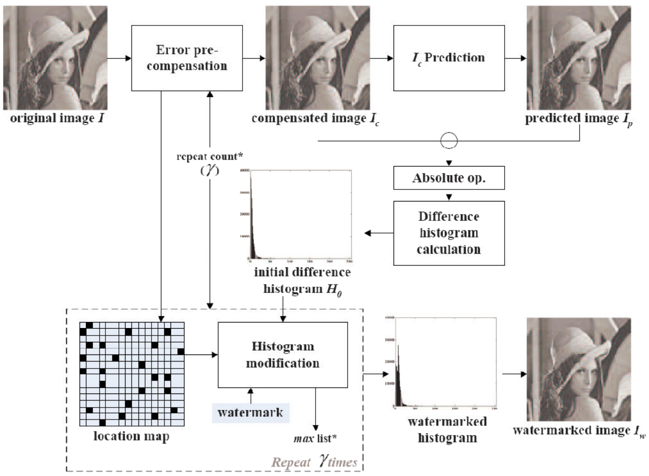

Fig. 2 Watermark embedding procedure.

An interpolated image for I_c, namely, I_p, is generated. I_p is then subtracted from I_c to calculate an absolute difference image δ_0 and obtain a histogram H_0. Note that when the difference histogram is calculated, the fixed pixels from the prediction phase are excluded.

The following procedure is repeated as long as $k \le \gamma$. The initial inputs for H_k and δ_k are H_0 and δ_0, respectively. We find the maximum bin of H_k and its position max_k. The histogram is then shifted to obtain an empty bin as the embedding space. The shifted histogram H_{k+1} is calculated as follows:

$$\begin{aligned} H_s(i,j) &= \begin{cases} H_k(i,j)+1, & \text{if } H_k(i,j)>max_k \\ H_k(i,j), & \text{otherwise} \end{cases} \\ \Leftrightarrow \quad \delta_s(i,j) &= \begin{cases} \delta_k(i,j)+1, & \text{if } \delta_k(i,j)>max_k \\ \delta_k(i,j), & \text{otherwise} \end{cases} \end{aligned}$$

It is apparent that max_0 is 0 for γ=1. However, max_k may not be equal to zero, depending on the characteristics of the input image and γ. Therefore, the information on max_k for each k is embedded into the image as side information.

After creating the empty bin, the watermark w is embedded by modifying H_k. The pixels that correspond to the difference value of max_k are considered as the embedding positions. We modify only those difference values of the pixels for which the watermark bit to be embedded is 1. The watermark embedding can be formulated as follows:

$$\begin{aligned} H_{k+1}(i,j) &= \begin{cases} H_s(i,j)+1, & \text{if } H_s(i,j)=max_k \text{ and } w(n)=1 \\ H_s(i,j), & \text{otherwise} \end{cases} \\ \Leftrightarrow \quad \delta_{k+1}(i,j) &= \begin{cases} \delta_s(i,j)+1, & \text{if } \delta_s(i,j)=max_k \text{ and } w(n)=1 \\ \delta_s(i,j), & \text{otherwise} \end{cases} \end{aligned}$$

where n is an index in the watermark sequence. At this point, it is checked whether k is equal to γ-1. If yes, this step ends. Otherwise, k increases by 1 and this step is repeated.

The watermarked image I_w is obtained from the modified-difference image δ_γ by adding δ_γ or -δ_γ to I_p. Since the difference histogram is calculated using the absolute pixel differences, the sign should be considered as follows:

$$I_w(i,j)=\begin{cases} I_p(i,j)-\delta_\gamma(i,j), & \text{if } I_c(i,j)\ge I_p(i,j) \\ I_p(i,j)+\delta_\gamma(i,j), & \text{if } I_c(i,j)<I_p(i,j) \end{cases}$$

D.Detection Algorithm

Fig. 3 shows the entire detection procedure. The detection method involves a four-step procedure. For watermark detection, γ and the information on max_k for $1 \le k \le \gamma$ are required. These information can be represented by less than one byte. Therefore, it is compressed by the LSB compression and embedded into the watermarked image.

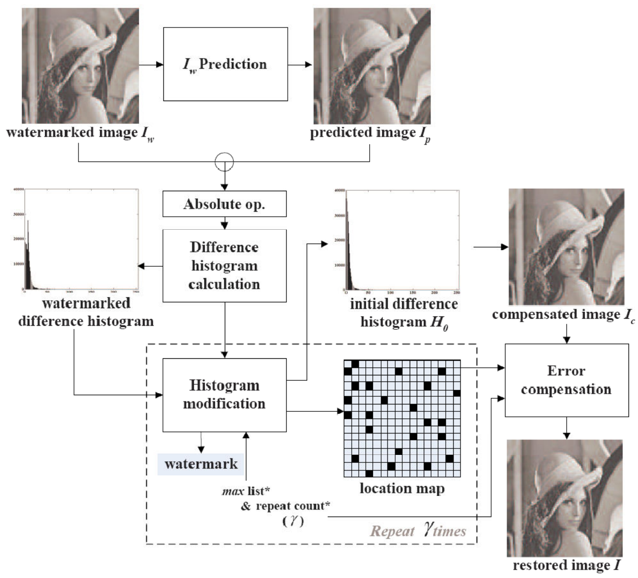

Fig. 3 Watermark detection procedure

When the input is I_w, I_p is generated. Note that the pixels of I_w remain unchanged. Hence, the image interpolated using I_w is identical to that obtained from I_c. I_p is then subtracted from I_w to calculate δ_γ and obtain H_γ.

By utilizing the information on γ and max_k for $1 \le k \le \gamma$, the watermark bits are extracted and the histogram is restored. The following procedure is repeated as long as $k \ge 1$. The initial inputs for H_k and δ_k are H_γ and δ_γ, respectively. Note that this procedure is performed in the reverse order.

The watermark extraction is modeled as follows:

$$w(n)=\begin{cases}0, & \text{if } \delta_k(i,j)=max_k \\ 1, & \text{if } \delta_k(i,j)=max_k+1\end{cases}$$

After watermark extraction, the watermark is eliminated from the watermarked difference image, which can be expressed by the following equation:

$$\begin{aligned} H_s(i,j) &= \begin{cases} H_k(i,j)-1, & \text{if } H_k(i,j)=max_k+1 \\ H_k(i,j), & \text{otherwise}\end{cases} \\ \Leftrightarrow \quad \delta_s(i,j) &= \begin{cases} \delta_k(i,j)-1, & \text{if } \delta_k(i,j)=max_k+1 \\ \delta_k(i,j), & \text{otherwise}\end{cases} \end{aligned}$$

The shifted difference histogram is then restored to its former form.

$$\begin{aligned} H_{k-1}(i,j) &= \begin{cases} H_s(i,j)+1, & \text{if } H_s(i,j)>max_k \\ H_s(i,j), & \text{otherwise}\end{cases} \\ \Leftrightarrow \quad \delta_{k+1}(i,j) &= \begin{cases} \delta_s(i,j)-1, & \text{if } \delta_s(i,j)>max_k \\ \delta_s(i,j), & \text{otherwise}\end{cases} \end{aligned}$$

At this point, it is checked whether k is equal to 1. If yes, this step ends. Otherwise, k is decreased by 1 and this step is repeated.

Since the original difference image is obtained, I_c, which is not watermarked, can be obtained. As in the embedding step, the restoration step is modeled as

$$I_c(i,j)=\begin{cases} I_p(i,j)-\delta_0(i,j), & \text{if } I_w(i,j)\ge I_p(i,j) \\ I_p(i,j)+\delta_0(i,j), & \text{if } I_w(i,j)<I_p(i,j)\end{cases}$$

The restored image I_c is not the original image. Rather, it is the compensated image. The original image can be restored by using γ and the location map μ. Since the location map is transmitted as a part of the watermark and γ is transferred as side information, the original image is obtained by the following operation:

$$I(i,j)=\begin{cases} I_c(i,j)+\gamma, & \text{if } \mu(i,j)=1 \text{ and } I_c(i,j)\ge 128 \\ I_c(i,j)-\gamma, & \text{if } \mu(i,j)=1 \text{ and } I_c(i,j)<128 \\ I_c(i,j), & \text{otherwise}\end{cases}$$

IV. Experimental Results

We evaluate the capacity and visual quality achieved by the proposed method using eight grayscale images of 512x512 pixels. To assess the performance of our method, the capacity is measured in bits per pixel (bpp) and is defined as the amount of data that can be hidden, excluding overhead information.

A. Performance Evaluation

Since the capacity is determined by γ, the performance of our algorithm is analyzed in terms of the capacity and visual quality, achieved for different repeat counts ranged 1 to 9. Fig. 4 shows the performance obtained by using the proposed method for different repeat counts.

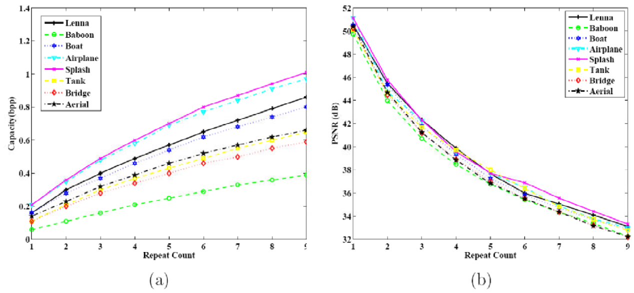

Fig. 4 Comparison of the performance with different repeat counts: (a) the capacity and (b) the visual quality

Fig. 4 (a) shows plots of the capacity for different repeat counts. It can be observed that as the repeat count increases, the capacity gradually increases but the rate of increase decreases. This is because as the embedding process is repeated, the interpolation error increases, and hence, the maximum frequency of the histogram is smaller than that in the previous embedding process. Fig. 4 (b) shows plots of visual quality for different repeat counts. The figure shows that as the repeat count increases, the visual quality gradually decreases because of the increased capacity. In the case of the method developed by Kim et al. [3], when the number of repetitions of embedding increased, the change in performance depended on the characteristics of the images. This was because they exploited the differences between the pixels of sub-sampled images. On the other hand, in the case of our proposed method, the qualities of the eight images degraded gradually, irrespective of the image characteristics. Thus, it is shown that the performance of the proposed method based on the interpolation error is stable, irrespective of the characteristics of the input images.

Fig. 5 shows some examples of the original images and marked images at various embedding capacities. As shown in the figures, the visual quality of the marked images is satisfactory at a high capacity.

B. Performance Comparison with Other Algorithms

The proposed algorithm was compared with the RS method [6], G-LSB method [7], DE-based method [10], and the Kim et al. method [3] in terms of the capacity and image quality using the lena, baboon, boat, and airplane images, as shown in Fig. 6. The capacity of the DE-based and G-LSB methods was relatively high, given the allowable PSNR value, whereas the capacity of the RS method was low compared to those of the other methods. The performance of the histogram-based Kim et al. method was better than those of the other methods, but not as good as that of our proposed method.

Experimental results show that the proposed algorithm yields a high capacity and high visual quality and outperforms other reversible watermarking methods.

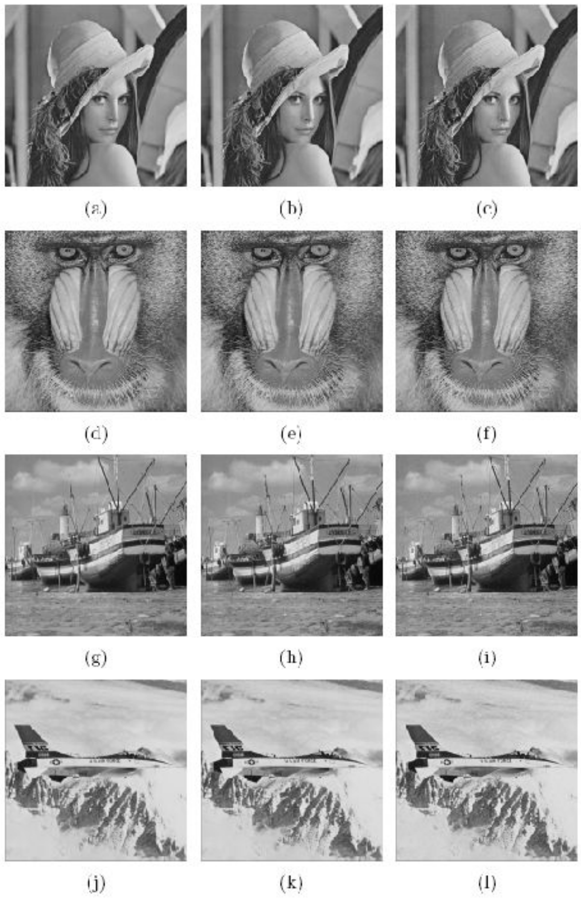

Fig. 5 Original and marked images: (a) lenna, (b) 42.35 dB with 0.40 bpp, (c) 35.09 dB with 0.73 bpp, (d) baboon, (e) 40.72 dB with 0.16 bpp, (f) 34.34 dB with 0.33 bpp, (g) boat, (h) 41.62 dB with 0.29 bpp, (i) 34.75 dB with 0.53 bpp, (j) airplane, (k) 42.27 dB with 0.48 bpp, and (l) 34.88 dB with 0.83 bpp

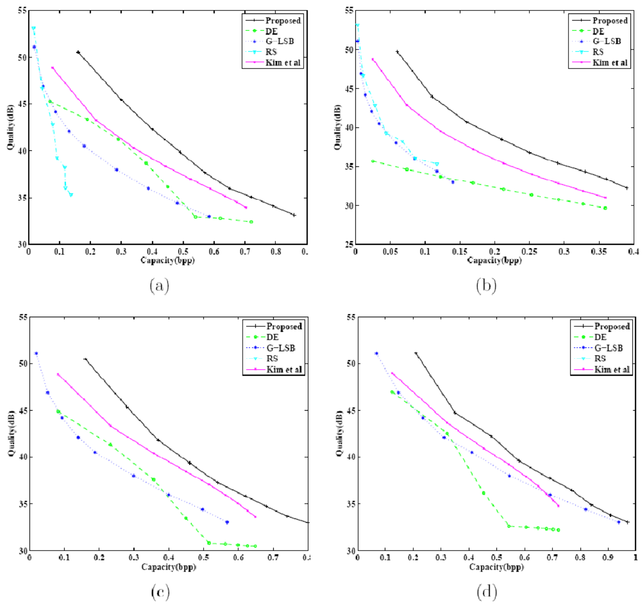

Fig. 6 Comparison of capacity vs visual quality: DE [10], G-LSB [7], RS [6], and Kim et al. [3]. (a) lenna, (b) baboon, (c) boat, (d) airplane

V. Conclusions

With the rapid spread of digital content, copyright protection has become critical. Digital watermarking techniques are essential for copyright protection. In particular, reversible watermarking has been introduced as a solution to satisfy the requirements of quality-sensitive applications such as medical, military, and artistic imaging. This paper proposed a reversible watermarking method that is based on histogram modification. We used the difference histogram for the original image and the image interpolated using the original image. The watermark was embedded and extracted by modifying the difference histogram. In comparison with the existing histogram-based methods, our proposed method achieved high capacity and high quality. Also, error pre-compensation prevents overflow and underflow. Our method can also be used for image authentication and tamper proofing.

Acknowledgment

This research project was supported by the 2013 Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2010-0007789).

References

[1] I. Cox, M. Miller, J. Bloom, J. Fridrich, T. Kalker, "Digital Watermarking and Steganography", Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2008.

[2] H. Sencar, M. Ramkumar, A. Akansu, "Watermarking Fundamentals and Applications", Elsevier Academic Press, UK, 2004.

[3] K.-S. Kim, M.-J. Lee, H.-Y. Lee, H.-K. Lee, "Reversible Watermarking Exploiting Spatial Correlation between Sub-Sampled Images", Pattern Recognition, Elsevier, vol.42, no.11, pp.3083-3096, 2009.

[4] R. Naskar, R.S. Chakraborty, "Performance of Reversible Digital Image Watermarking under Error-Prone Data Communication: A Simulation-Based Study", IET Image Processing, IET, vol.6, no.6, pp.723-737, 2012.

[5] F. Mintzer, J. Lotspiech, N. Morimoto, "Safeguarding Digital Library Contents and Users: Digital Watermarking", D-Lib Magazine, 1997.

[6] J. Fridrich, J. Goljan, R. Du, "Invertible Authentication", In Proceedings of the SPIE Security and Watermarking of Multimedia Contents III, San Jose, CA, vol.4314, pp.197-208, 2001.

[7] M. U. Celik, G. Sharma, A. M. Tekalp, E. Saber, "Lossless Generalized-LSB Data Embedding", IEEE Trans. on Image Processing, IEEE, vol.14, no.2, pp.253-266, 2005.

[8] B. Yang, M. Schmucker, C.B.W. Funk, S. Sun, "Integer DCT-Based Reversible Watermarking for Images Using Compounding Technique", In Proceedings of the SPIE Security, Steganography, and Watermarking of Multimedia Contents VI, San Jose, CA, vol.5306, pp.405-415, 2004.

[9] S. Lee, C.D. Yoo, T. Kalker, "Reversible Image Watermarking Based On Integer-To-Integer Wavelet Transform", IEEE Trans. on Information Forensics and Security, IEEE, vol.2, no.3, pp.321-330, 2007.

[10] J. Tian, "Reversible Data Embedding Using a Difference Expansion", IEEE Trans. on Circuits and Systems for Video Technology, IEEE, vol.13, no.8, pp.890-896, 2003.

[11] A.M. Alattar, "Reversible Watermark Using the Difference Expansion of a Generalized Integer Transform", IEEE Trans. on Image Processing, IEEE, vol.13, no.8, pp.1147-1156, 2004.

[12] L. Kamstra, H.J.A.M. Heijmans, "Reversible Data Embedding into Images Using Wavelet Techniques and Sorting", IEEE Trans. on Image Processing, IEEE, vol.14, no.12, pp.2082-2090, 2005.

[13] D. Coltuc, "Improved Embedding for Prediction-Based Reversible Watermarking", IEEE Trans. on Information Forensics and Security, IEEE, vol.6(3), pp.873-882, 2011.

[14] Y. Hu, H.-K. Lee, K. Chen, J. Li, "Difference Expansion Based Reversible Watermarking Using Two Embedding Directions", IEEE Trans. on Multimedia, IEEE, vol.10, pp.1500-1512, 2008.

[15] Z. Ni, Y.-Q. Shi, N. Ansari, W. Su, "Reversible Watermarking", IEEE Trans. on Circuits and Systems for Video Technology, IEEE, vol.16, no.3, pp.354-362, 2006.
[16] B. Lin, W. Tai, C. Chang, "Multilevel Reversible Watermarking Based On Histogram Modification of Difference Images", Pattern Recognition, Elsevier, vol.41, pp.3582-3591, 2008.
[17] P. Tsai, Y.-C. Hu, H.-L. Yeh, "Reversible Image Hiding Scheme Using Predictive Coding and Histogram Shifting", Signal Processing, Elsevier, vol.89, no.6, pp.1129-1143, 2009.
[18] JBIG-KIT Lossless Image Compression Library, URL http://www.cl.cam.ac.uk/mgk25/ jbigkit/