

Journal of Electronic Imaging

SPIDigitalLibrary.org/jei

Robust image watermarking based on luminance modification

Narong Mettripun
Thumrongrat Amornraksa
Edward J. Delp



Robust image watermarking based on luminance modification

Narong Mettripun

Thumrongrat Amornraksa

King Mongkut's University of Technology Thonburi

Faculty of Engineering

Computer Engineering Department

126 Pracha-Uthit Road, Thung khru

Bangkok 10140, Thailand

E-mail: mettripun_n@hotmail.com

Edward J. Delp

Purdue University

School of Electrical and Computer Engineering

465 Northwestern Avenue

West Lafayette, Indiana 47907-2035

Abstract. A robust image watermarking technique based on a modification of the luminance component of the host color image is described. Three methods are proposed in the watermarking scheme in order to improve its performance in terms of accuracy of the extracted watermark and robustness of the embedded watermark. These methods are a new approach to watermark embedding in the luminance component of a host image and a new original image prediction technique in the watermark extraction process. A set of experiments is carried out to verify our proposed methods. The experimental results show significant improvements gained from our proposed watermarking scheme compared to previous existing schemes. The results also show enhanced robustness of the embedded watermark against various types of attacks. Our proposed watermarking scheme can be used for both color and gray scale images. © The Authors. Published by SPIE under a Creative Commons Attribution 3.0 Unported License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: 10.1117/1.JEI.22.3.033009]

1 Introduction

In the digital era, various types of media, including audio, video, and images can easily be duplicated and distributed without permission from original owner/creator. This is undesirable because the consequences of such actions may discourage the owner/creator from developing future work. One possible solution to this problem is the use of digital watermarking to discourage people from making and distributing unauthorized copies of digital media.¹ Image watermarking is a method used to embed information imperceptibly (the watermark) into a host image before public distribution. The degradation of a watermarked image must be unnoticeable by the observer of the image. The embedded watermark must be robust against both unintentional and

intentional attacks while being extractable so the watermark can be “read”.^{2,3} A watermarking method used for image distribution should also be capable of blind detection so that the watermark can be extracted without the original image.⁴

At present, various image watermarking schemes have been proposed and shown to be robust against various types of attacks. Several of them embed a watermark within the transform domain of the host image^{5,6,7,8} so that the embedded watermark can survive most compression schemes, such as JPEG and JPEG2000. There are also some studies demonstrated that such approaches are robust against geometrical attacks, e.g., cropping.^{4,7,8} However, most methods suffer from low capacity in that few watermark bits can be added to the host image.⁹ A simple and fast approach, based on spatial domain watermarking, was thus considered as an alternative. It was shown in many studies that the embedded watermark can survive most of the geometrical attacks while simultaneously providing considerably high watermark capacity. For example, the blind watermarking method proposed by Verma et al. in 2007 described watermark embedding by modifying the pixel values in the blue (B) component of a color image.¹⁰ Note that the blue component was modified because the human visual system (HVS) is least sensitive to blue.¹ In the scheme, a 3×3 pixel block from the predefined 8×8 pixel image was modified in such a way that watermark extraction could be achieved by comparing the average intensity of subsets of pixels of the 8×8 block. An error-correcting code was used with the embedded bits in order to enhance the performance. However, this scheme provided a small watermark capacity of 2500 bits/ 512×512 color image pixels. In 2008, another color image watermarking scheme based on linear discriminant analysis (LDA) was proposed, where all three color channels [i.e., red (R), green (G) and blue (B)] were used to carry a watermark in the form of a binary logo image. A trained LDA machine was used for watermark extraction.¹¹ The scheme provided a small watermark

Paper 12387 received Sep. 19, 2012; revised manuscript received May 7, 2013; accepted for publication Jun. 24, 2013; published online Aug. 12, 2013.

capacity of 800 bits/ $512 \times 512 \times 3$ color image pixels and also required a reference watermark to train the LDA in the extraction process. In 2009, a localized image watermarking resistant to geometric attacks was proposed by Li and Guo.¹² In their scheme, the watermark was embedded into all local invariant regions repeatedly in the spatial domain of a color image and could be extracted from the distorted image directly with the help of an odd–even bit detector. Since the embedding positions were restricted to be within the local invariant regions in order to guard against geometric attacks (such as rotation, scaling, and translation), only a small number of bits (e.g., 16 bits) could be embedded into a 512×512 gray level image. Recently, Hussein proposed a nonblind watermarking scheme based on log-average luminance¹³ whereby some 8×8 pixel blocks, which were chosen spirally from the center of the embedding image and had a log-average luminance greater than or equal to the log-average luminance of the entire image, were used for watermark embedding. However, in this scheme, apart from using an inconvenient nonblind approach, modifying the luminance components of host image significantly degraded the visual image quality which could be perceived by a human observer. Although the author avoided this drawback by allowing only 16 blocks to be modified, the performance of the method was limited to having a small watermark capacity, as only 1024 bits were embedded into 512×512 color image pixels.

A blind watermarking scheme based on the modification of image pixels enabling a large number of embedding bits was first proposed by Kutter et al.,¹⁴ where watermark embedding was performed by modifying the blue component of color image pixels, and watermark extraction was achieved by using a prediction method based on a linear combination of pixel values in a cross-shaped neighborhood around the embedded pixels. For this method, 512×512 bits could be embedded into 512×512 color image pixels. The method was experimentally observed to be robust against various types of image attacks, including geometrical attacks. The extraction performance was improved by introducing a Gaussian pixel-weighting mask into the embedding process and employing a linear combination of all nearby pixel values around the embedded pixel.¹⁵ However, if the number of watermark bits “1” and “–1” around the embedding pixel was not equal or balanced,¹⁶ the summation of those watermark bits would result in a large value, which directly affects the accuracy of the original pixel prediction step in the watermark extraction process, and the probability of extracting the watermark correctly would decrease. Such circumstances can frequently occur when a watermark that is to be embedded consists of recognizable patterns. The extraction probability also decreases when the host image is a very detailed image, that is when two nearby pixel values are substantially different. A similar concept of watermark embedding was also presented in Ref. 17, where the proposed perceptual mark was based on the adaptive least square (LS) prediction error sequence of the host image and claimed to match well with the properties of the HVS. Together with the new blind detection scheme based on an efficient prewhitening process and a correlation-based detector, their proposed mark exhibited impressive performance and the watermark capacity in the scheme was comparable to Ref. 15. However, the watermark embedding in

the luminance component greatly degraded the perceptual quality of the watermarked image when compared with watermark embedding in the blue component at the same watermark strength. Watermark embedding in luminance translates to watermark embedding in all three color components, i.e., RGB. Thus, the resultant image quality will be degraded in accordance with the changes in each R, G, and B component. Based on the weaknesses in Ref. 15, three different improving techniques were proposed in Ref. 16. These techniques included balancing the watermark bits around the embedding pixel, tuning the strength of embedding watermark in accordance with nearby luminance components and reducing the effect caused by substantially different values between the nearby watermarked component and the center one in the prediction area. A different approach for improving the performance of this watermarking scheme was also presented in Ref. 18, where the watermark is embedded into the chrominance components of YC_bC_r color space that have less variation value. Although it achieved a better extraction performance, the accuracy of the extracted watermark still suffered from most compression schemes, e.g., a low quality watermark was obtained after applying JPEG compression.

We present in this article a new watermarking scheme based on the modification of image pixels in order to improve the accuracy of the extracted watermark and the robustness of the embedded watermark, as proposed in Refs. 14–18, especially against image compression standards. Three different methods are proposed to improve the overall performance. First, we propose a new watermark embedding method in the luminance component of the host image instead of the color component to avoid high lossy compression used in many image compression methods. This approach is usually overlooked because the quality of the host image will be severely dropped. Second, we reduce the number of watermark bits to be embedded, based on discrete wavelet transforms (DWTs) without decreasing the watermark image size in order to reduce the modifying number of the luminance components in the host image. Third, we propose a new watermark extraction method based on the prediction of original pixel from the weighted watermarked components in order to suit the high variation value of the watermarked luminance components. The performance of all three proposed methods is evaluated and compared with the previous watermarking schemes. The next section describes the proposed methods including our watermarking scheme. Section 3 presents the experimental settings and the performance of our proposed scheme, compared to the others. The conclusion is finally included in Sec. 4.

2 Proposed Methods

2.1 Watermark Embedding Based on Luminance Modification

We first consider embedding a watermark into a luminance component rather than in color components. This is because, in general, image compression methods strongly decrease the chrominance quality of a color image through subsampling processes.¹⁹ The watermark embedded in the luminance component should therefore be more robust against image compression than those in the color components. YC_bC_r color space is one of the most well known models

and widely used to present images. We thus choose the component Y in this color space, which is separately encoded, for watermark embedding. Recall that in YC_bC_r color space, for example, Y represents the luminance component of a color image, whereas C_b and C_r represent the blue and red chrominance components, respectively.²⁰ In addition, an image in RGB color space can be converted to YC_bC_r , or vice versa, by the following equations:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (1)$$

and

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.164 & 0 & 1.596 \\ 1.164 & -0.392 & -0.813 \\ 1.164 & 2.017 & 0 \end{bmatrix} \begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} + \begin{bmatrix} -222.921 \\ 135.576 \\ -276.836 \end{bmatrix}. \quad (2)$$

However, since the HVS is very sensitive to changes in the luminance component, changing values of Y undoubtedly cause a more severe effect on perception than changes in the color and/or chrominance components.

One efficient solution we consider here is to decrease the number of embedding bits in order to reduce the effect of the embedded watermark bits on the Y component while simultaneously improving the quality of the watermarked image. Nevertheless, this solution must neither affect the size of the embedded watermark nor excessively degrade its quality. Figure 1 shows the zoomed version of the host and watermarked images "Lena" for the B and Y components.

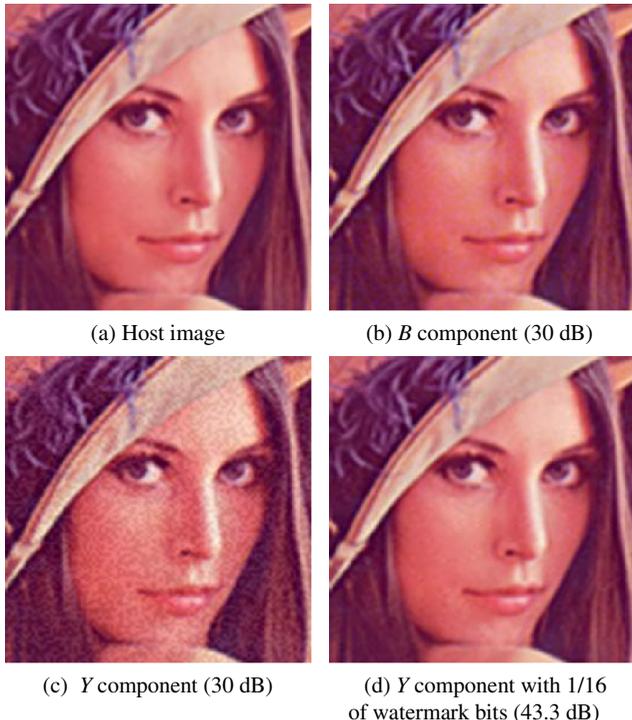


Fig. 1 Zoomed version of original and watermarked images of "Lena."

Figure 1(d) demonstrates the result obtained from embedding only 1/16 of the watermark bits, with the same strength as used in Fig. 1(c), into the same host image. Note that the watermarking scheme in Ref. 16 was used in this test and the quality of watermarked image was controlled to achieve a peak signal to noise ratio (PSNR) of 30 dB. PSNR for watermarking embedding in the Y component is given by:

$$\text{PSNR (dB)} = 20 \log \frac{255\sqrt{3MN}}{\sqrt{\sum_{i=1}^M \sum_{j=1}^N [Y'(i,j) - Y(i,j)]^2}}, \quad (3)$$

where $Y(i,j)$ and $Y'(i,j)$ are the original and watermarked Y component at coordinates (i,j) , whereas M and N are the numbers of the row and column in the image, respectively. In case of the B component, $Y(i,j)$ and $Y'(i,j)$ are replaced by $B(i,j)$ and $B'(i,j)$, respectively. Note that PSNR is an objective quality measure that is not consistent with HVS.

Figure 1 shows that at the same PSNR, the quality of watermarked image in Y component, Fig. 1(c), was perceptually poorer than that in B component, Fig. 1(b). However, when the number of watermark bits in the same Y component was reduced to 1/16 of the original number, with the same strength, the change in quality is not very visible [see Fig. 1(d)], and the PSNR was increased to 43.3 dB.

2.2 Watermark Preparation/Reconstruction Based on DWT

To accomplish the embedding in the Y component without affecting the watermark excessively, we develop a new watermark consisting of three processing steps. The first two are based on the two-dimensional (2-D) DWT and are used to reduce the size of the embedding watermark and to enlarge the size of the extracted watermark to its original dimensions. The last processing step is based on image denoising and is used to diminish the negative consequences of propagation error. The 2D-DWT of function $f(x,y)$ of size $M \times N$ is defined as follows²¹:

$$W_\varphi(j_0, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \varphi_{j_0, m, n}(x,y), \quad (4)$$

$$W_\psi^i(j, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \psi_{j, m, n}^i(x,y), \quad (5)$$

where j_0 is an arbitrary starting scale and the coefficients $W_\varphi(j_0, m, n)$ define an approximation of $f(x,y)$ at scale j_0 . The coefficients $W_\psi^i(j, m, n)$ add horizontal (H), vertical (V), and diagonal (D) details for scales $j \geq j_0$. We normally let $j_0 = 0$ and select $M = N = 2^J$ so that $j = 0, 1, 2, \dots, J-1$ and $m, n = 0, 1, 2, \dots, 2^j - 1$. $\varphi_{j, m, n}(x,y)$, and $\psi_{j, m, n}^i(x,y)$ in Eqs. (4) and (5) are defined as follows:

$$\varphi_{j, m, n}(x,y) = 2^{j/2} \varphi(2^j x - m, 2^j y - n), \quad (6)$$

$$\psi_{j, m, n}^i(x,y) = 2^{j/2} \psi(2^j x - m, 2^j y - n), \quad (7)$$

where index i identifies the directional wavelets as follows:

$$\psi^H(x,y) = \psi(x)\varphi(y), \quad (8)$$

$$\psi^V(x, y) = \varphi(x)\psi(y), \quad (9)$$

$$\psi^D(x, y) = \psi(x)\psi(y). \quad (10)$$

Equation (6) identifies the scaling function as follows:

$$\varphi(x, y) = \varphi(x)\varphi(y). \quad (11)$$

In this article, we use the unit-height, unit-width scaling function and the Haar wavelet²¹ function for 2-D DWT in order to decompose an image to four quarter size subimages, namely, W_φ , W_ψ^H , W_ψ^V , and W_ψ^D . Both the functions are given in Eqs. (12) and (13):

$$\varphi(x) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & \text{elsewhere} \end{cases} \quad (12)$$

and

$$\psi(x) = \begin{cases} 1 & 0 \leq x < 0.5 \\ -1 & 0.5 \leq x < 1. \\ 0 & \text{elsewhere} \end{cases} \quad (13)$$

Note that the decomposition process can be used again to the approximation of the image to obtain another set of four subband images. The resulting decomposition of the image after doing the 2-D DWT two times is illustrated in Fig. 2.

It should be noted that $W_\varphi(j+1, m, n)$ can be reconstructed from $W_\varphi(j, m, n)$ and $W_\psi^i(j, m, n)$, and $W_\varphi(j+2, m, n)$ from $W_\varphi(j+1, m, n)$ and $W_\psi^i(j+1, m, n)$, via the inverse DWT.

$$f(x, y) = \frac{1}{\sqrt{MN}} \sum_m \sum_n W_\varphi(j_0, m, n) \varphi_{j_0, m, n}(x, y) + \frac{1}{\sqrt{MN}} \sum_{i=H,V,D} \sum_{j=j_0}^{\infty} \sum_m \sum_n W_\psi^i(j_0, m, n) \psi_{j, m, n}^i(x, y). \quad (14)$$

To use the 2-D DWT to construct our watermark, the watermark image $I_w(i, j) \in \{0, 1\}$ with the same size of the host image is first created from a black-and-white recognizable pattern and then decomposed two times using 2-D DWT to obtain seven subimages. Next, each coefficient co_w in $W_\varphi(j, m, n)$ (see Fig. 2) is modified to obtain a two-level value by the following:

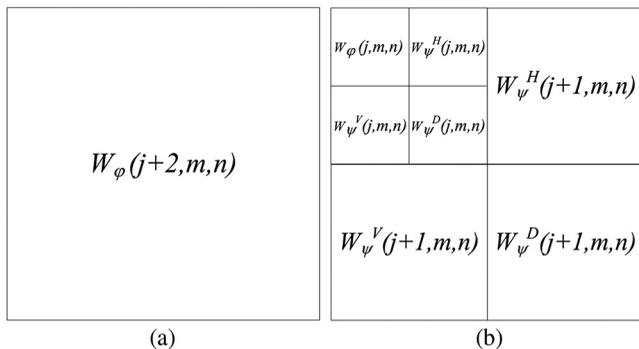


Fig. 2 Examples of (a) original image (b) subimages after taking two 2-D discrete wavelet transform (DWT) decompositions.

$$co_{w_mod}(i, j) = \begin{cases} 1 & co_w(i, j) \geq 2 \\ 0 & \text{elsewhere} \end{cases}, \quad (15)$$

where co_{w_mod} is the modified coefficient in $W_\varphi(j, m, n)$. Note that since the value of co_w varies from 0 to 4, we use the value 2 at the midpoint of this range as a threshold to convert $co_w(i, j)$ to $co_{w_mod}(i, j) \in \{0, 1\}$. The resulting 1/16 size of the subimage is used as the watermark and contains only 6.25% of the watermark bits from the original version. The seven subimages obtained after two 2-D DWT decompositions to the two-color, black-and-white image “Scout Logo” are illustrated in Fig. 3.

When the extracted watermark co'_{w_mod} is recovered, the second step is applied in order to reconstruct co'_{w_mod} to its original size. That is, each coefficient is modified in accordance with the following equation to obtain a two-level value co'_{w_new} .

$$co'_{w_new}(i, j) = \begin{cases} 4 & co'_{w_mod}(i, j) = 1 \\ 0 & \text{elsewhere} \end{cases}. \quad (16)$$

The new subimage $W_{\varphi_new}(j, m, n)$ containing co'_{w_new} together with the new recreated subimages $W_{\psi_new}^i(j, m, n)$ and $W_{\psi_new}^i(j+1, m, n)$ containing all zero coefficients are inverse transformed to reconstruct the watermark image I'_w in its original size. Note that the values of lower and upper bounds, i.e., 0 and 4, are used in Eq. (16) because, based on observations, the quality of the reconstructed image using these two values is closer to the original image than with any other value. However, in the second step, if an erroneous coefficient in $W_{\varphi_new}(j, m, n)$ occurs, this will lead to a group of 16 erroneous pixels in I'_w . Hence, in the last step, a 5×5 pixel denoising filter with the following property is applied to I'_w to reduce the effect of the propagation error. Note that this image denoising technique works in such a way that the output pixel value depends on the majority of I'_w being within an area of 5×5 pixels.

$$I''_w(i, j) = \begin{cases} 1 & \text{when } \left[\sum_{m=-2}^2 \sum_{n=-2}^2 I'_w(i+m, j+n) \right] \geq 13 \\ 0 & \text{elsewhere} \end{cases}. \quad (17)$$

The example of subimage co_{w_mod} from the image Scout Logo together with six new recreated subimages and its enlarged version is shown in Fig. 4(a) and 4(b), while the

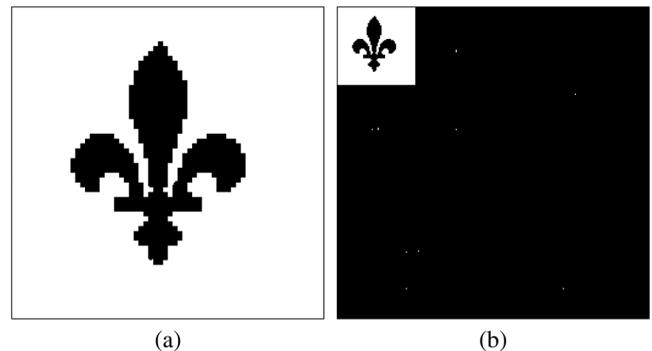


Fig. 3 Example of (a) original image “Scout Logo” and (b) its corresponding subimages after two 2-D-DWT decompositions.

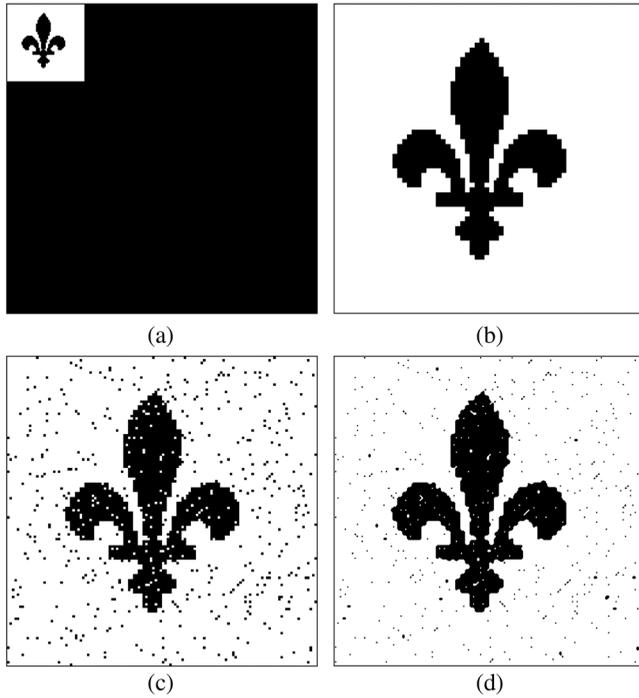


Fig. 4 (a) Example of subimage co_{w_mod} and six new recreated sub-images and (b) its enlarged version (c) the extracted watermark images Scout Logo before the denoising filter and (d) after the denoising filter.

extracted watermark images Scout Logo before and after the denoising filter are shown in Fig. 4(c) and 4(d), respectively.

Based on the above watermark construction, a new watermarking scheme based on luminance modification is proposed. The block diagram showing steps in watermark embedding is illustrated in Fig. 5. The steps in the watermark embedding process are as follows: First, after obtaining the reduced size of I_w by the 2-D DWT decompositions, co_{w_mod} is XORed with a pseudorandom bit stream of the same length generated by a key-based stream cipher in order to obtain a balanced set of bits around the embedding component and thus providing security for the embedded watermark. That is, without the secret key, no one can reproduce the same pseudorandom bit stream used in the

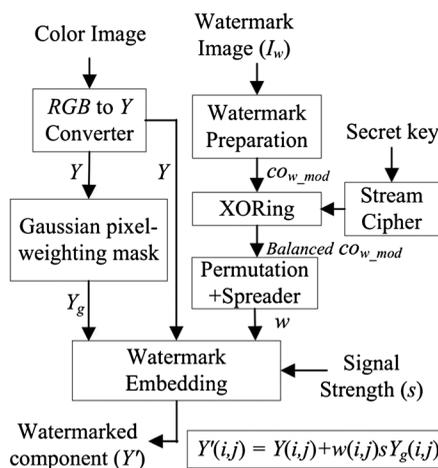


Fig. 5 Block diagram of the proposed watermark embedding.

embedding processing, and as a result would be unable to recover the embedded watermark. The bit positions of the result are then permuted and spread in accordance with the uniform distribution to disperse groups of 0 and 1 bits over the entire embedding area. In practice, all k watermark bits are first permuted and spread randomly based on the uniform distribution over $16k$ pixel positions. Finally, the 0 bits are converted into -1 so that the watermark to be embedded becomes $w(i, j) \in \{-1, 1\}$. Note that the remaining $(15/16)$ pixels of the host image remains unchanged.

To watermark a host color image, the luminance component of the host image at coordinate (i, j) is pseudorandomly modified by using addition or subtraction, depending on the value of $w(i, j)$, the watermark strength s , and the luminance component of the embedding pixel $Y(i, j)$. The tuning factor, s , is included here in order to control the overall quality of the watermarked image. In practice, s is a constant value used to achieve an expected PSNR and may be different depending on the host image. According to Eq. (1), the luminance component is determined by $Y(i, j) = 0.299R(i, j) + 0.587G(i, j) + 0.114B(i, j)$. Note that no luminance component of the host image is embedded two times, so that only $1/16$ of the entire luminance component is modified. The watermarked luminance component $Y'(i, j)$ can be represented by:

$$Y'(i, j) = Y(i, j) + w(i, j)sY_g(i, j), \quad (18)$$

where $Y_g(i, j)$ is the modified luminance value of the 3×3 pixel block obtained from the Gaussian pixel-weighting mask,¹⁵ which is considered as an HVS-based tuning factor for watermark strength. In practice, s must be carefully selected to obtain the best trade-off between imperceptibility and robustness.

2.3 Original Pixel Prediction Based on Weighted Components

The block diagram showing steps in the proposed watermark extraction process is illustrated in Fig. 6.

From the figure, an embedded watermark can be recovered based on two assumptions. First, we assume that any pixel value within an image is close to its surrounding neighbors so that a pixel value at a given coordinate (i, j) can be estimated using the average of the values of its nearby pixels. Hence, a prediction of $Y(i, j)$, which we denote as $Y''(i, j)$, is determined from the nearby watermarked components around (i, j) as follows:

$$Y''(i, j) = \frac{1}{8} \left\{ \left[\sum_{m=-1}^1 \sum_{n=-1}^1 Y'(i+m, j+n) \right] - Y'(i, j) \right\}. \quad (19)$$

Second, we assume that the summation of w around (i, j) is close to zero so that the embedded bit at (i, j) can be estimated by the following equation:

$$w'(i, j) = Y'(i, j) - Y''(i, j). \quad (20)$$

It was shown in Ref. 16 that replacing a surrounding neighbor around (i, j) that most differed from $Y''(i, j)$ by $Y'(i, j)$ itself can help improve the accuracy of Y'' . Since $w'(i, j)$ can be either positive or negative, the zero value is set as its threshold, and its sign is used to estimate the value

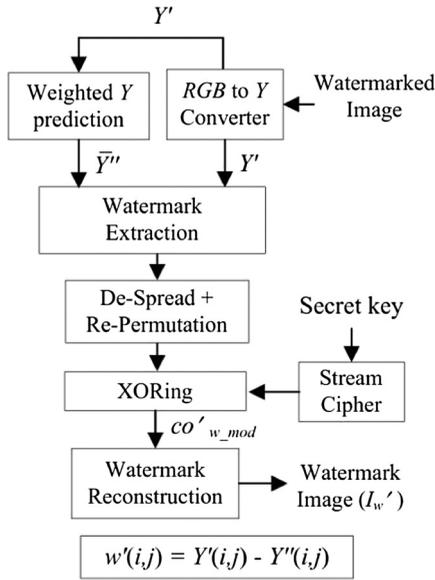


Fig. 6 Block diagram of the proposed watermark extraction process.

of $w(i, j)$. That is, if $w'(i, j)$ is positive (or negative), $w(i, j)$ is estimated as 1 (or -1). Note that the magnitude of $w'(i, j)$ reflects the confidence level of estimating $w(i, j)$. Last, the -1 bit of $w'(i, j)$ is converted into 0, and the result is despread, repermuted and then XORed with the same pseudorandom bit stream used in the embedding process to obtain the recovered, black-and-white image $I'_w(i, j) \in \{0, 1\}$. Note that the same pseudorandom bit stream can be reproduced if the watermark detector knows the secret key.

From the above two assumptions, the accuracy of the extracted watermark depends mainly on the variation of the image pixels. For example, the two neighbor pixels

having highly different values have a high chance of obtaining an error prediction for $w'(i, j)$. In fact, the variation of the Y component after being watermarked using the above scheme always increases, and hence, unavoidably causes a lower accuracy on watermark extraction. To enhance the performance of $w'(i, j)$ estimation of the Y component, we consider using a new prediction technique for $Y(i, j)$, taking into account the different values between two nearby components, i.e., the center and its neighbor. That is, instead of using the true value of the neighbor component around (i, j) in the prediction process, we first apply a weighting factor to every neighbor component around (i, j) , so that all neighbor components get closer to the center pixel. Conceptually, the weighting factor is determined based on the different values between the predicting component and its neighbors. Since, as mentioned earlier, a component value at coordinates (i, j) is assumed to be predicted from its neighbors, and the neighbor component value should be close to the predicting one. Also, since the range of values for the Y component varies from 16 to 235 and the different values between the two components can be varied from 0 to 219, the weighting factor is applied directly to the nearby component in accordance with the difference between that component and the predicting one. Based on this concept, the weighted neighbor component $\bar{Y}'(i, j)$ around $Y'(i, j)$ of an area 3×3 pixels can be represented by the following equation:

$$\bar{Y}'(i + m, j + n) = Y'(i + m, j + n) + \alpha[Y'(i, j) - Y'(i + m, j + n)], \quad (21)$$

where α is a constant value used to adjust the weighted component, m and $n = -1, 0, 1$. Finally, a new prediction of $Y(i, j)$, which we denote as $\bar{Y}''(i, j)$, is given by:

Table 1 Differences between five image watermarking schemes.

Scheme	Host and watermark images	Embedding method	Prediction method
Hussein's work denoted by $(Lu - Log, Y, org)$	- Host color image - Use black and white logo with the same size as host image	- Use Log average luminance value - Embed in Y component chosen spirally from the centre of the embedding image	No prediction method, and need original host image as reference
Kutter's work denoted by $(Lu, B, 4n)$	- Host color image - Use black and white logo with the same size as host image	- Use luminance value from each embedding pixel only - Embed in all blue components without XORing operation	Use cross-shape neighborhood (four watermarked components)
Karybal's work denoted by $(Lu - LS, Y, 8n)$	- Both color and gray scale images can be used as host image - Use black and white logo with the same size as host image	- Use spatial perceptual mask based on the adaptive LS prediction error of host image - Embed in all Y component with XORing operation	Use eight surrounding neighborhood (eight watermarked components)
Amornraksa's work denoted by $(Lu - G, B, 7 + 1n)$	- Host color image - Use black and white logo with the same size as host image	- Use luminance value weighted from the embedding pixel and its nearby pixels - Embed in all Blue components with XORing operation	Use seven surrounding neighborhood with its centre (eight watermarked components)
Proposed method denoted by $(Lu - G, Y/16, w8n)$	- Both color and gray scale images can be used as host image - Use black and white logo with the same size as host image	- Use luminance value weighted from the embedding pixel and its nearby pixels - Embed in only 1/16 of Y components with XORing operation	Use weighted value from surrounding neighborhood (eight watermarked components)

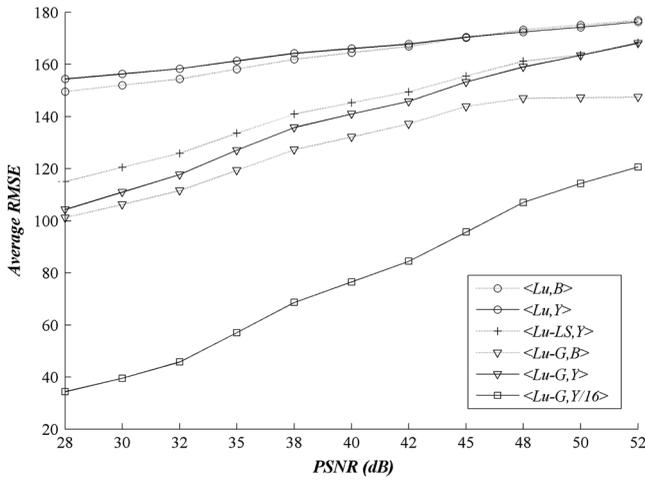


Fig. 7 Comparison of average root mean square error (RMSE) from different embedding methods at various peak signal to noise ratios (PSNRs).

$$\bar{Y}''(i, j) = \frac{1}{8} \left\{ \left[\sum_{m=-1}^1 \sum_{n=-1}^1 \bar{Y}'(i+m, j+n) \right] - \bar{Y}'(i, j) \right\}. \quad (22)$$

Note that $\bar{Y}'(i, j) = Y'(i, j)$, and $w'(i, j)$ is now obtained by:

$$w'(i, j) = Y'(i, j) - \bar{Y}''(i, j). \quad (23)$$

The differences between the proposed watermarking scheme and the previous equivalent schemes are summarized in Table 1. Note that, in comparison, we concentrated on the watermarking scheme that can embed the two-color watermark image having the same size as the original host image only.

3 Experimental Results

In all the experiments, sixteen 256×256 pixel color images having various characteristics, namely, “Lena,” “Airplane,” “Fish,” “Pepper,” “Tower,” “Baboon,” “House,” “Bird,” “Always running,” “A water trick,” “Couple,” “Golden Gate,” “Sail boat on lake,” “San Francisco,” “Splash,” and

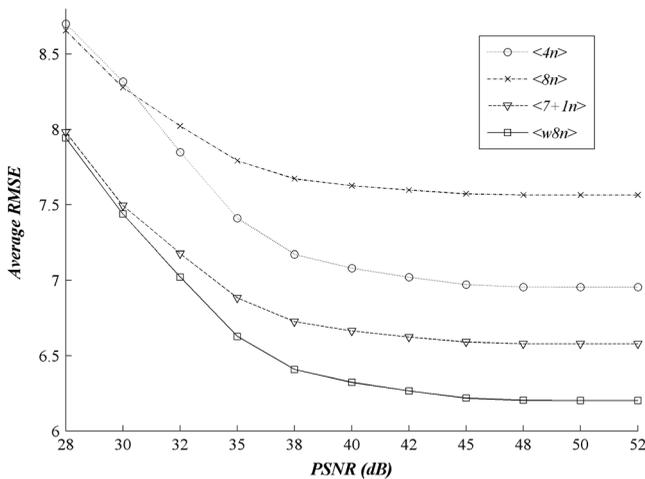


Fig. 8 Comparison of average RMSE from different prediction methods at various PSNRs.

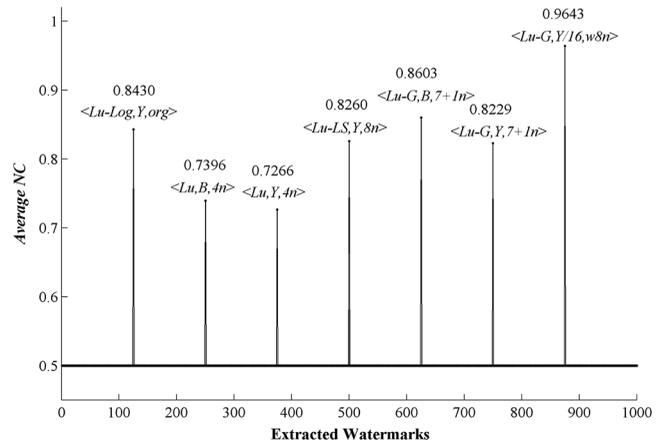


Fig. 9 Comparison of normalized correlation (NC) from 1000 watermarks.

“Tree” were used as original host images. Most of them were taken from Refs. 22 and 23. A black-and-white image of the same size as the host image Scout Logo was created and used as the watermark. To obtain a fair comparison between different watermarking schemes, the embedding parameters used in each scheme were adjusted until the quality of watermarked images reached quality image at PSNR of 35 dB.²⁴ When the watermark was extracted, its accuracy was evaluated by a metric known as the normalized correlation (NC). The robustness of the embedded watermark was also evaluated by the NC. The NC is a similarity measurement between two different signals, which is given as follows:

$$NC = \frac{\sum_{i=1}^M \sum_{j=1}^N I_w(i, j) I'_w(i, j)}{\sqrt{\sum_{i=1}^M \sum_{j=1}^N I_w(i, j)^2} \sqrt{\sum_{i=1}^M \sum_{j=1}^N I'_w(i, j)^2}}. \quad (24)$$

Normally, when two different versions of watermark are compared, the value of NC varies from 0 to 1, provided that each comparing watermark contains at least one component representing the value of 1. Note that the value of $NC = 1$ implies that two compared signals are identical. Also, the

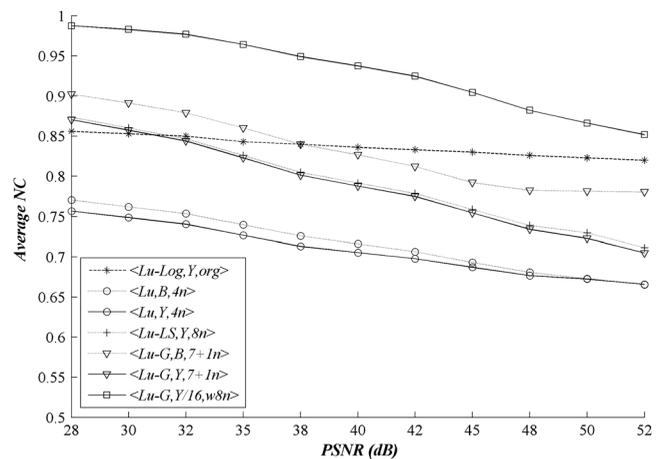


Fig. 10 Comparison of average NC obtained from different watermarking schemes at various PSNRs from different watermarking schemes at various PSNRs.

higher the NC is, the more accurate the extracted watermark will be. Apart from using the NC, the quality of the extracted watermark may be evaluated without comparing it to the original version. Since the watermark image contains recognizable patterns and/or logos, its quality may be judged from the intelligibility of its content. In this article, we mainly used the NC to evaluate the performance of the watermarking schemes though we sometimes used human observers to rapidly validate the extracted watermark.

For the experiments, we first explored the impact of the proposed watermark embedding method and the proposed original image prediction technique separately, before employing them in our watermarking scheme. We then evaluated and compared the performance of our scheme with

the previous schemes under the same circumstance, i.e., embedding a black-and-white image into a color image of the same size. Finally, we evaluated and compared the robustness of the seven watermarking schemes against various types of attacks including JPEG-based compression schemes. Two of them were in fact adapted from the blue component embedding method in Kutter's and Amornraksa's schemes to the Y component and denoted by $\langle Lu, Y, 4n \rangle$ and $\langle Lu - G, Y, 7 + 1n \rangle$, respectively.

3.1 Impacts of the Proposed Methods

To demonstrate that the proposed watermark constructed for the Y component helped to improve the accuracy of the

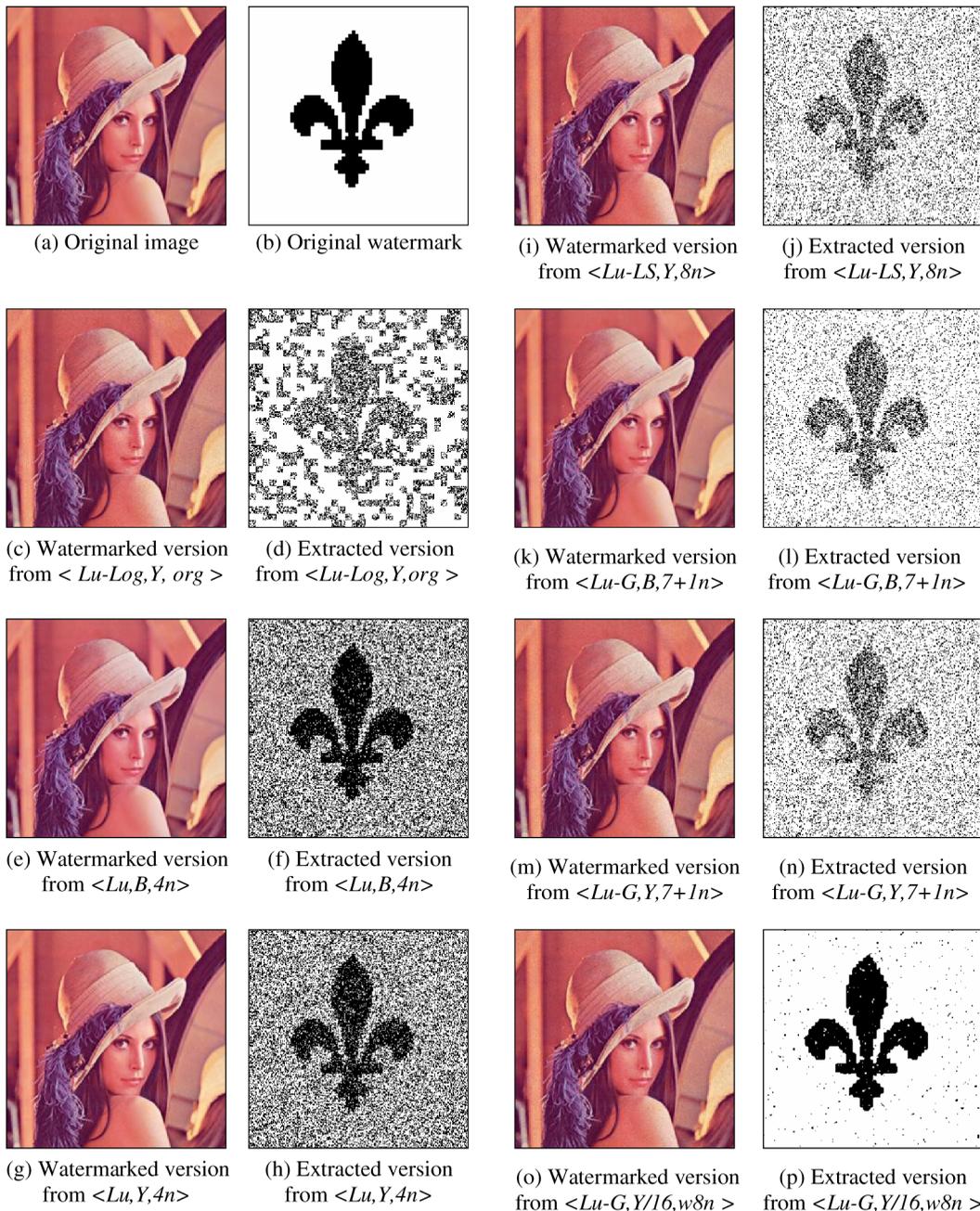


Fig. 11 Examples of watermarked image and its extracted watermark.

Table 2 Comparison of the average wPSNR at PSNR of 35 ± 0.01 dB.

Scheme	$\langle Lu - Log, Y, org \rangle$	$\langle Lu, B, 4n \rangle$	$\langle Lu, Y, 4n \rangle$	$\langle Lu - LS, Y, 8n \rangle$	$\langle Lu - G, B, 7 + 1n \rangle$	$\langle Lu - G, Y, 7 + 1n \rangle$	$\langle Lu - G, Y / 16, w8n \rangle$
Average wPSNR (dB)	37.4598	37.9030	37.9657	37.9127	37.7418	37.8430	37.8590
Difference (dB)	0.3992	-0.0439	-0.1066	-0.0537	0.1173	0.0160	0

extracted watermark, the root mean square error (RMSE) between the extracted and original watermarks, that is, w' and w , was measured to observe the performance of the proposed method versus existing methods. Note that a smaller value of RMSE indicates a lesser difference between two components. The results in terms of average RMSE at various PSNR from the watermarking scheme in Ref. 16 with different embedding methods and channels discussed above are shown in Fig. 7. In the figure, we denote the embedding methods of Kutter, Karybali, Amornraksa, and the proposed one, as described in Table 1, by $\langle Lu, B \rangle$, $\langle Lu, Y \rangle$, $\langle Lu - LS, Y \rangle$, $\langle Lu - G, B \rangle$, $\langle Lu - G, Y \rangle$, and

$\langle Lu - G, Y / 16 \rangle$, respectively. The proposed method achieved the highest accuracy with respect to the extracted watermark as compared to the other methods.

We then demonstrate that the quality of the predicted image obtained from the weighting-based prediction method is closer to the original image than that obtained from other existing methods. Again, we measured the RMSE between the predicted and original components for every embedding position in order to observe the difference between various prediction methods. For instance, in the B component, RMSE is computed from B'' and B , while in the Y component, from Y'' and Y . Note that in this situation, a smaller

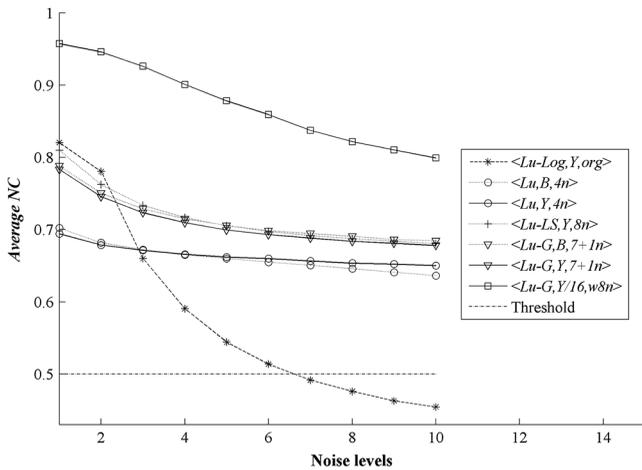


Fig. 12 Stirmark benchmark: random noise addition (normalized ranges between 0 and 100).

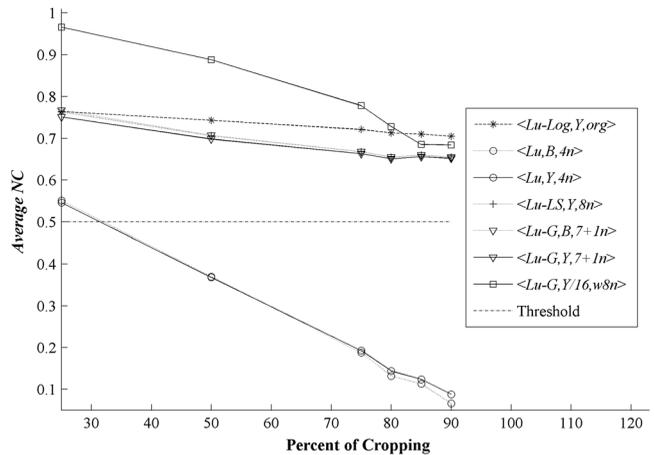


Fig. 14 Stirmark benchmark: cropping.

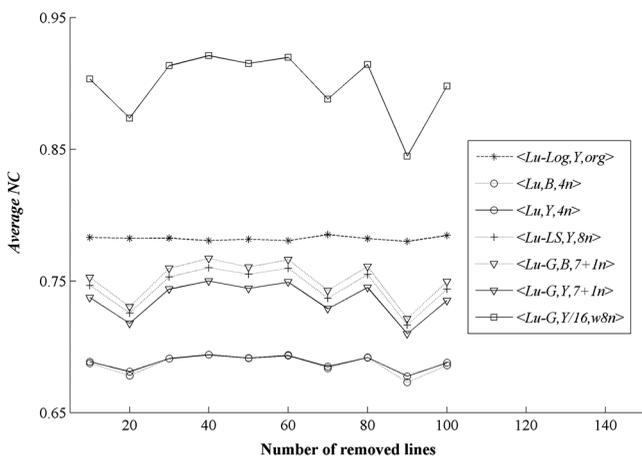


Fig. 13 Stirmark benchmark: line removal.

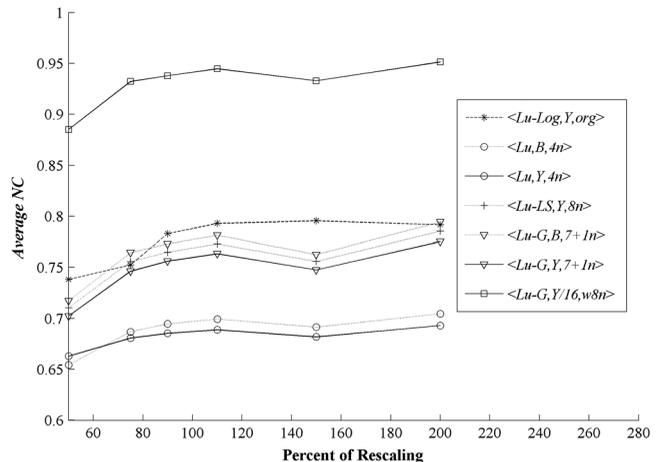


Fig. 15 Stirmark benchmark: rescaling.

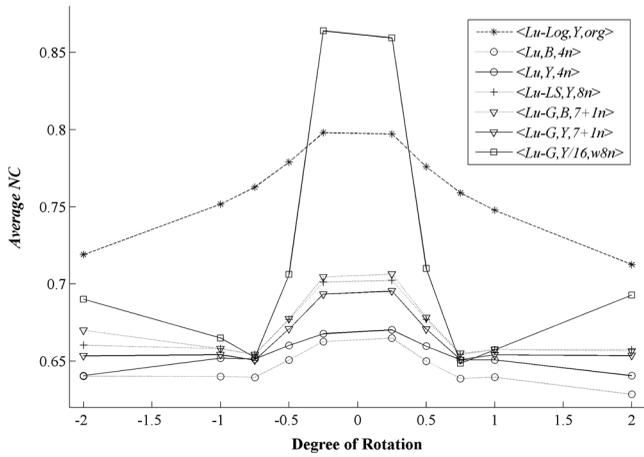


Fig. 16 Stirmark benchmark: image rotation.

value of RMSE indicates a better prediction of the original component. The results of RMSE averaged from all host images based on various PSNR from the watermarking scheme in Ref. 16 with different prediction methods described in Table 1 are presented and compared in Fig. 8. From the figure, we denote the prediction methods of Kutler, Karyali, Amornraksa, and the proposed one by $\langle 4n \rangle$, $\langle 8n \rangle$,

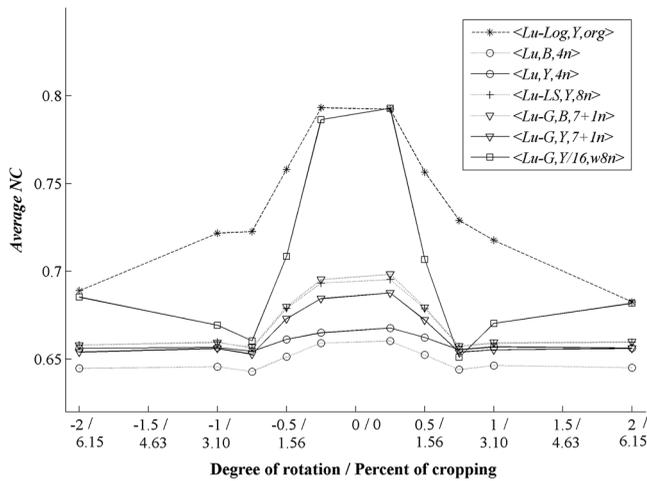


Fig. 17 Stirmark benchmark: rotation and cropping.

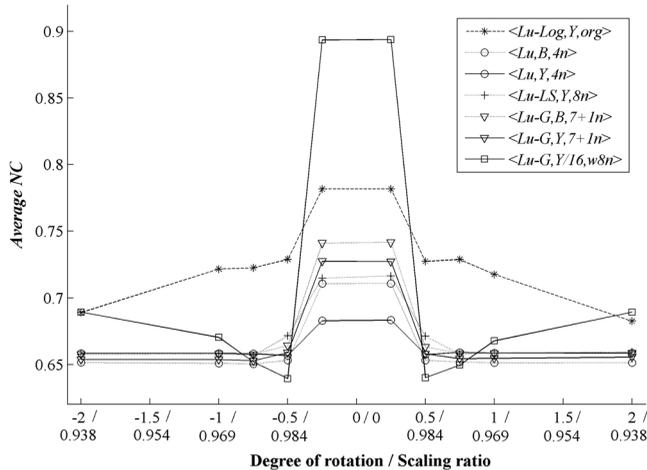


Fig. 18 Stirmark benchmark: rotation and scaling.

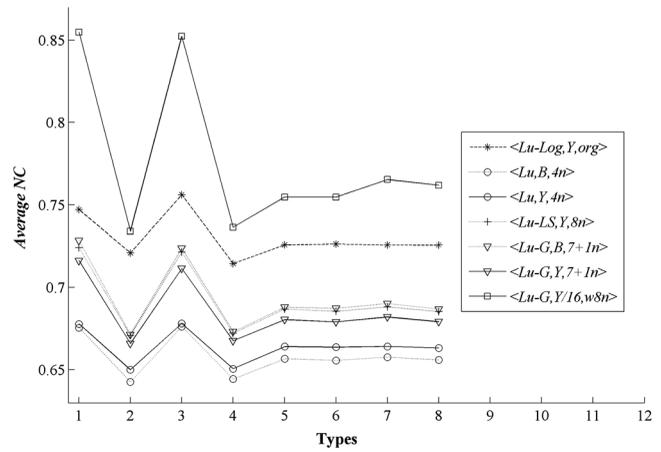


Fig. 19 Stirmark benchmark: affine transform.

$\langle 7 + 1n \rangle$, and $\langle w8n \rangle$, respectively. The results verified that our prediction method obtained the highest quality predicted image at all PSNR values. Note that in the case of pixel prediction at the edge of image, the value of the missing pixels was replaced by the nearest pixel. Note that the Hussein scheme¹³ was not compared here because it does not need the prediction step.

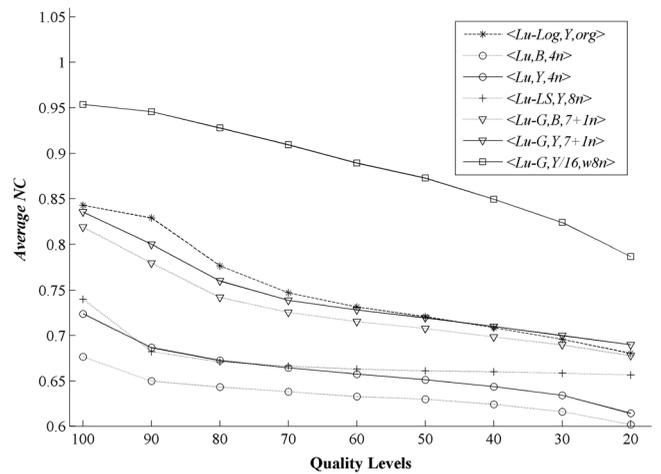


Fig. 20 Average NC values at various JPEG image qualities.

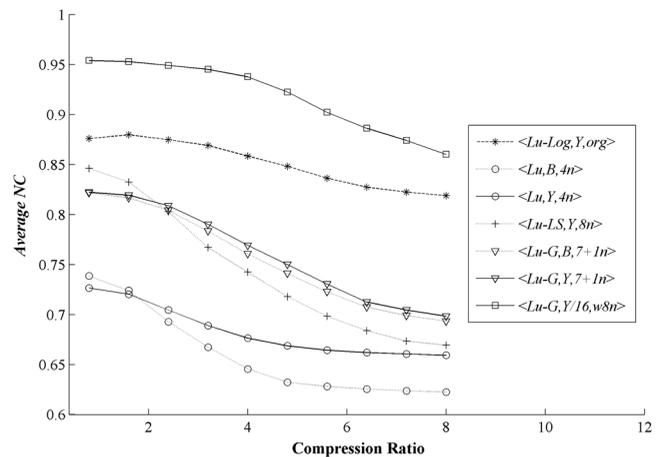


Fig. 21 Average NC values at various JPEG2000 image qualities.

3.2 Performance Comparison

First, we needed to identify the value of NC used to differentiate the extracted watermark from the fake one. To accomplish this, we deployed a watermark counterfeit attack by computing the average value of NC of the watermark extracted from all watermarked testing images and comparing the results from the seven watermarking schemes with 993 different watermarks. In the experiments, the quality of all watermarked images was controlled to achieve 35 dB, and the value of α was set to 0.475 to obtain the best prediction performance. Note that α was obtained experimentally by a full-search approach. That is, by searching the value of α that gave the highest NC value, on average, from

all testing images. In the experiments, the value of α was varied in step of 0.1, from 0 to 1. According to the results shown in Fig. 9, the average NC value between the extracted genuine watermark and the other 993 watermarks was approximately 0.5. Hence, if the values of NC for an extracted watermark was lower than 0.5, the watermark could be presumed to be a fake. This threshold may be used to indicate the absence of an embedded watermark as well because the value of NC for a valid watermark after the XORing step was equivalent to that obtained by using a pseudorandom bit stream (7 genuine and 993 random watermarks).

Now, we compared the performance of the seven watermarking schemes. The results in terms of average NC at

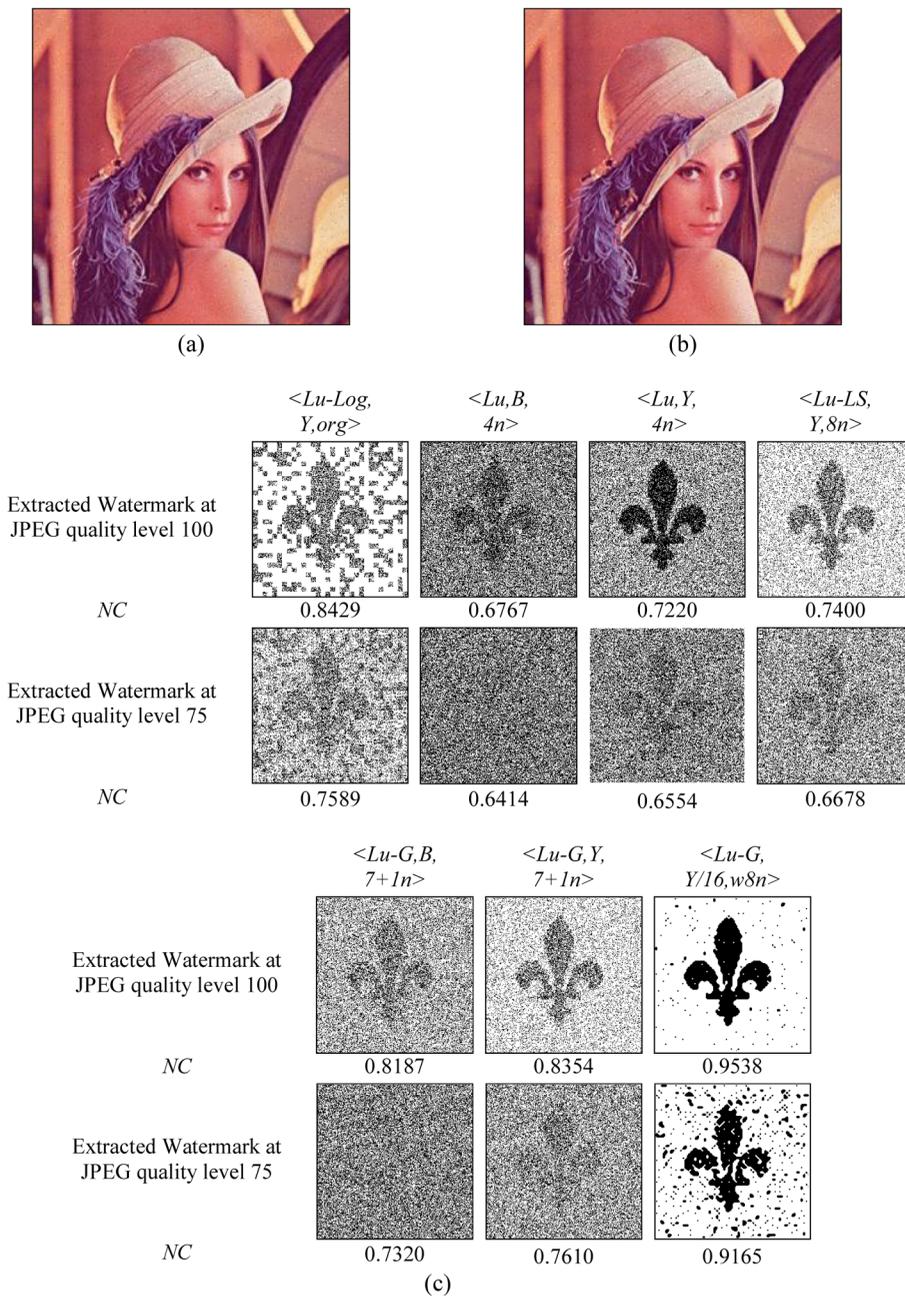


Fig. 22 The JPEG compressed watermarked image at (a) 100% and (b) 75% quality factor. (c) Examples of the extracted watermarks from the seven watermarking schemes.

various PSNR are presented in Fig. 10. The proposed scheme outperforms the other schemes. It should be noted that the performance of $\langle Lu - \text{Log}, Y, \text{org} \rangle$ was not good even though it used the original host image to help extract the watermark. This is because the black and white logo with the same size as the host image was used in the experiments, and some image areas having too low log-average luminance were not used to carry watermark bits.

Examples of the original color image Lena, which included the two-color, black-and-white watermark Scout Logo, the watermarked image and the extracted watermark using the seven different schemes at PSNR of 35 dB are given in Fig. 11. The values of average NC obtained from $\langle Lu - \text{Log}, Y, \text{org} \rangle$, $\langle Lu, B, 4n \rangle$, $\langle Lu, Y, 4n \rangle$, $\langle Lu - \text{LS}, Y$,

$8n \rangle$, $\langle Lu - G, B, 7 + 1n \rangle$, $\langle Lu - G, Y, 7 + 1n \rangle$, and $\langle Lu - G, Y/16, w8n \rangle$ were 0.8430, 0.7396, 0.7226, 0.8260, 0.8603, 0.8229, and 0.9643, respectively.

Since embedding a watermark with the same strength to different components will result in different PSNRs, the watermarked images from different schemes and components were then fairly compared with another objective quality measure that matches well with the HVS properties, i.e., the weighted PSNR (wPSNR) taken from checkmark.²⁵ The wPSNR is an adaptation of PSNR that introduces different weights for the perceptually different image areas taking into account that the visibility of noise in flat image areas is higher than that in textures and edges.²⁶ The calculation of wPSNR is given by:

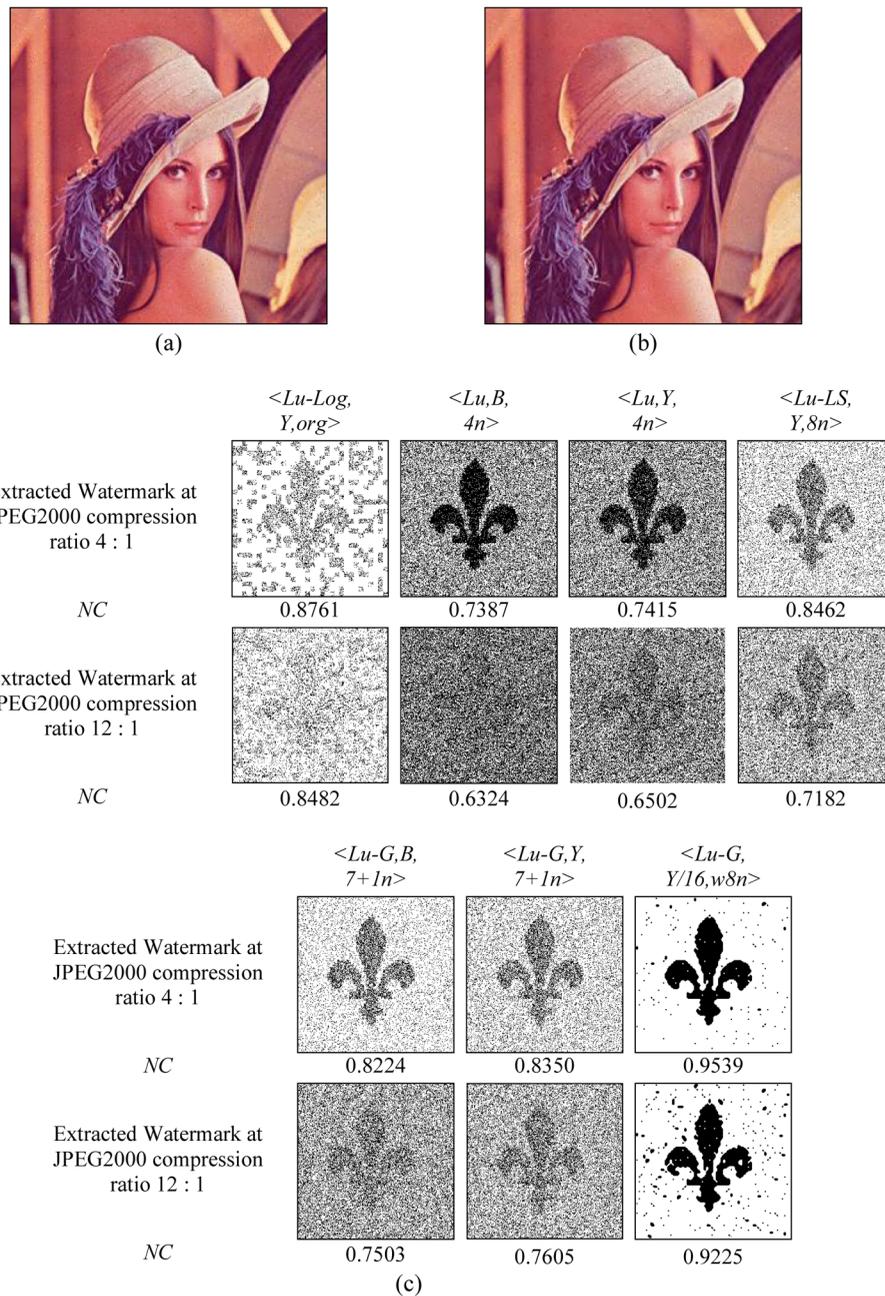


Fig. 23 The JPEG 2000 compressed watermarked image at compression ratio of (a) 4 : 1 and (b) 12 : 1. (c) Examples of the extracted watermarks from the seven watermarking schemes.

$$w\text{PSNR (dB)} = 20 \log$$

$$\times \frac{255\sqrt{3MN}}{\sqrt{\sum_{i=1}^M \sum_{j=1}^N \{N\text{VF}[Y'(i,j) - Y(i,j)]\}^2}}, \quad (25)$$

where NVF is noise visibility function which characterizes the local texture of the image and varies between 0 and 1, where it takes 1 for flat areas and 0 for highly textured regions.²⁶ In this experiment, the wPSNR values from all testing watermarked images at PSNR of 35 ± 0.01 dB were evaluated and compared. The results in terms of average wPSNR among the seven watermarking schemes are shown in Table 2. Obviously, the average wPSNR value from the proposed watermarking scheme was slightly lower than the three comparing schemes, i.e., $\langle \text{Lu}, B, 4n \rangle$, $\langle \text{Lu}, Y, 4n \rangle$, and $\langle \text{Lu} - \text{LS}, Y, 8n \rangle$.

3.3 Robustness Against Attacks

The various types of attacks were next implemented against the watermarked images using the Stirmark benchmark (version 4)^{27,28} and common image processing techniques. We then tried to extract the embedded watermark. After the attacks, if the size of the attacked image was different from its original version, we rescaled it to obtain the original size. In the case of a cropped image, we replaced the missing part(s) of the image by white pixels. It should be noted that the quality of all attacked images fell below 35 dB, depending on the type and strength of the attack. As demonstrated in Figs. 12–21 and Table 3, the average NC from the proposed scheme of the extracted watermark after being attacked was superior to the other schemes.

The examples of the watermarked image Lena using the proposed scheme after JPEG compression at 100% and 75% quality factor, which included the corresponding extracted watermark Scout Logo using the seven different schemes,

Table 3 Robustness comparison against various types of attack.

Type and strength of attack		Average normalized correlation (NC)						
		$\langle \text{Lu} - \text{Log}, Y, \text{org} \rangle$	$\langle \text{Lu}, B, 4n \rangle$	$\langle \text{Lu}, Y, 4n \rangle$	$\langle \text{Lu} - \text{LS}, Y, 8n \rangle$	$\langle \text{Lu} - G, B, 7 + 1n \rangle$	$\langle \text{Lu} - G, Y, 7 + 1n \rangle$	$\langle \text{Lu} - G, Y, 16, w8n \rangle$
Median filter	3 × 3 pixels	0.6796	0.6104	0.6450	0.6628	0.6685	0.6657	0.7107
Convolute filter	Smoothing	0.6997	0.4819	0.5726	0.7147	0.7091	0.7068	0.9079
	Sharpening	0.7118	0.7079	0.7095	0.8315	0.8424	0.8255	0.9732
Self similarities	type 1	0.7217	0.6926	0.6748	0.7813	0.7762	0.7263	0.8942
	type 2	0.7014	0.6180	0.7239	0.7576	0.7595	0.8179	0.9680
	type 3	0.7233	0.7384	0.6568	0.7624	0.7757	0.6719	0.7828
Small random distortions	0.95	0.7462	0.6310	0.6345	0.6571	0.6571	0.6539	0.7025
	1	0.7469	0.6298	0.6343	0.6577	0.6565	0.6535	0.7013
	1.05	0.7474	0.6288	0.6345	0.6578	0.6568	0.6540	0.7013
	1.1	0.7479	0.6285	0.6341	0.6564	0.6571	0.6537	0.7017
Latest small random distortions	0.95	0.7429	0.6356	0.6342	0.6575	0.6570	0.6539	0.7048
	1	0.7438	0.6349	0.6336	0.6574	0.6574	0.6539	0.7043
	1.05	0.7445	0.6345	0.6336	0.6572	0.6574	0.6531	0.7068
	1.1	0.7450	0.6342	0.6330	0.6574	0.6571	0.6532	0.7075
Brightness enchantment	+50	0.7371	0.6616	0.6710	0.7838	0.7911	0.7985	0.9224
	-50	0.7306	0.6947	0.6847	0.7976	0.8173	0.8203	0.9271
Contrast enchantment	+50	0.7588	0.6324	0.6443	0.7836	0.7828	0.7982	0.9205
	-50	0.7455	0.7197	0.7297	0.7985	0.8297	0.8302	0.9279

Table 4 Robustness comparison against watermark removal attack.

Version of recreated image "Lena"	NC			
	(Lu,B,4n)	(Lu - LS,Y,8n)	(Lu - G,B,7 + 1n)	(Lu - G,Y/16,w8n)
First round combination (35.18 dB)	0.7731	0.8739	0.9039	0.9858
Second round combination (31.27 dB)	0.7082	0.7415	0.8831	0.9718
Third round combination (29.42 dB)	0.4839	0.6182	0.7851	0.9566
Fourth round combination (28.28 dB)	0.5846	0.6769	0.7362	0.9309
Fifth round combination (27.50 dB)	0.4426	0.6415	0.7054	0.8997
Sixth round combination (26.90 dB)	0.5328	0.6544	0.6918	0.8732
Seventh round combination (26.43 dB)	0.4204	0.6462	0.6817	0.8599
Eighth round combination (26.04 dB)	0.4990	0.6502	0.6744	0.8512
Ninth round combination (25.71 dB)	0.4070	0.6451	0.6719	0.8406

are shown in Fig. 22, whereas similar examples after JPEG2000 with compression ratios of 4:1 and 12:1 and decompression layers of 5 are shown in Fig. 23.

Finally, we demonstrated the robustness of the embedded watermark against watermark removal. In this experiment, the predictions for the Y' (Y'') components as well as the C_b and C_r components were combined together to recreate an image without a watermark. The same process was applied to the resulting image several times with the aim of completely removing the embedded watermark. For example, the first-round combination was $Y'' + C_b + C_r$, the second round combination was $(Y'')'' + C_b + C_r$, and so on. The values of NC for the watermark extracted from different versions of the recreated image Lena and based on the four watermarking schemes are given in Table 4. Again the Hussein scheme¹³ was not included because it has no prediction step. The results obtained from the table confirmed that the embedded watermark still remained within all versions of the recreated image and could be reliably extracted. Also note that after the first round, the PSNR of the recreated image fell below 35 dB and became lower with each round it was recreated in.

4 Conclusions

We have presented a new image watermarking scheme based on luminance modification. The watermark was embedded into the luminance component of the host image without significant perceptible degradation. The luminance component prediction using the concept of a weighting factor was also employed to enhance the performance of the proposed watermarking scheme. The experimental results showed significant improvement in the proposed watermarking scheme in terms of accuracy of the extracted watermark and robustness of the embedded watermark, compared with the previous existing schemes, especially against two popular image compression methods, JPEG and JPEG2000.

For a practical system, other techniques such as error control coding or multiple embedding might be incorporated to

provide extra reliability for watermark extraction, provided the complexity in the new system is not too high and enough watermark bits can still be embedded.

Acknowledgments

This research work was supported by the Commission on Higher Education scholarship (CHE-PhD-SW-NEWU) granted to Mr. Narong Mettripun. The authors would like to sincerely thank Mr. Suwat Tachaphetpiboon and Miss Kharitha Thongkor for their fruitful discussions.

References

1. I. J. Cox et al., *Digital Watermarking, and Steganography*, 2nd Ed., Morgan Kaufmann Publishers, Burlington, Massachusetts (2008).
2. J. J. K. O'Ruanaidh, W. J. Dowling, and F. M. Boland "Watermarking digital images for copyright protection," in *IEE Proc. Vision Image Signal Process.*, **143**(4), 250–256 (1996).
3. M. Barni and F. Bartolini, *Watermarking Systems Engineering Enabling Digital Assets Security and Other Application*, 1st Ed., Marcel Dekker, Inc., New York (2004).
4. F. Y. Shih, *Digital Watermarking And Steganography Fundamentals and Techniques*, 1st Ed., CRC Press, Boca Raton, Florida (2007).
5. C. V. Serdean et al., "Wavelet and multiwavelet watermarking," *IET Image Process.* **1**(2), 223–230 (2007).
6. S. Agreste et al., "An image adaptive, wavelet-based watermarking of digital images," *J. Comput. Appl. Math.* **210**(1–2), 13–21 (2007).
7. A. Poljicak, L. Mandic, and D. Agic, "Discrete Fourier transform-based watermarking method with an optimal implementation radius," *J. Electron. Imaging* **20**(3), 033008 (2011).
8. X. Kang, J. H Zhang, and W. Zeng, "Efficient general print-scanning resilient data hiding based on uniform log-polar mapping," *IEEE Trans. Inform. Forensics Secur.* **5**(1), 1–12 (2010).
9. J. Seitz, *Digital Watermarking for Digital Media*, Information Science Publishers, Hershey, Pennsylvania (2005).
10. B. Verma, D. P. Agarwal, and S. Jain, "Spatial domain robust blind watermarking scheme for color image," *Asian J. Inform. Technol.* **6**(4), 430–435 (2007).
11. Y. G. Fu and R. Shen, "Color image watermarking scheme based on linear discriminant analysis," *Comput. Stand. Interfaces* **30**(3), 115–120 (2008).
12. L. Li and B. Guo, "Localized image watermarking in spatial domain resistant to geometric attacks," *AEU Int. J. Electron. Commun.* **63**(2), 123–131 (2009).
13. J. A. Hussein, "Spatial domain watermarking scheme for colored images based on log-average luminance," *J. Comput.* **2**(1), 100–103 (2010).

14. M. Kutter, F. Jordan, and F. Bossen, "Digital watermarking of color images using amplitude modulation," *J. Electron. Imaging* **7**(2), 326–332 (1998).
15. R. Puernpan and T. Amornraksa, "Gaussian pixel weighting marks in amplitude modulation of color image watermarking," in *Proc. Int. Conf. IEEE ISSPA*, pp. 194–197, Kuala Lumpur, Malaysia (2001).
16. T. Amornraksa and K. Jantawongwilai, "Enhanced images watermarking based on amplitude modulation," *Image and Vision Comput.* **24**(2), 111–119 (2006).
17. I. G. Karybali and K. Berberidis, "Efficient spatial image watermarking via new perceptual masking and blind detection schemes," *IEEE Trans. Inform. Forensics Secur.* **1**(2), 256–274 (2006).
18. K. Surachat and T. Amornraksa, "Pixel-wise based digital watermarking in $YCbCr$ color space," in *Proc. Int. Conf. PCM 2009*, pp. 1293–1299, Springer-Verlag, Berlin, Heidelberg, Germany (2009).
19. P. Symes, *Video Compression Demystified*, 1st Ed., McGraw-Hill, New York (2001).
20. Recommendation BT.601-5, "Studio encoding parameters of digital television for standard 4:3, and side-screen 16:9 aspect ratios," Section 11B: Digital Television (1995).
21. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd Ed., Prentice Hall, Upper Saddle River, New Jersey (2002).
22. The USC-SIPI Image Database, "Signal and image processing institute," University of Southern California, <http://sipi.usc.edu/database> (June 2010).
23. IRTC Viewing and Voting (Stills), "Results for September-October 1998," Topic: Water, <http://www.irtc.org/stills/1998-10-31.html> (June 2010).
24. M. Li, S. Narayanan, and R. Poovendran, "Tracing medical images using multi-band watermarks," in *Proc. IEEE Int. Conf. Engineering in Medicine and Biology Society (IEMBS '04)*, pp. 3233–3236, San Francisco (2004).
25. CheckMark, Vision Group University of Geneva, <http://cvml.unige.ch/ResearchProjects/Watermarking/Checkmark/> (May 2013).
26. S. Voloshynovskiy et al., "Attack modelling: towards a second generation benchmark," *Signal Process.* **81**(6), 1177–1214 (2001).
27. F. A. P. Petitcolas, "Watermarking schemes evaluation," *IEEE Signal Process. Mag.* **17**(5), 58–64 (2000).
28. F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Attacks on copy-right marking systems," in *Proc. 2nd Int. Workshop on Information Hiding (IH'98)*, pp. 218–238, Springer-Verlag, London (1998).

Department of Electrical and Computer Engineering, Purdue University, USA.



Thumrongrat Amornraksa received MSc and PhD degrees from University of Surrey, England, in 1996 and 1999, respectively. He is currently an associate professor in the Computer Engineering Department, King Mongkut's University of Technology Thonburi (KMUTT). His research interests are digital image processing and digital watermarking.



Edward J. Delp received his BSEE and MS degrees from the University of Cincinnati, and a PhD degree from Purdue University. In 2002, he received an Honorary Doctor of Technology from the Tampere University of Technology in Tampere, Finland. He is currently The Charles William Harrison Distinguished Professor of Electrical and Computer Engineering and Professor of Biomedical Engineering and Professor of Psychological Sciences (Courtesy) His research interests include image and video compression, multimedia security, medical imaging, multimedia systems, communication, and information theory. He is a Fellow of the IEEE, a Fellow of the SPIE, a Fellow of the Society for Imaging Science and Technology (IS&T), and a Fellow of the American Institute of Medical and Biological Engineering. In 2008 Dr. Delp received the Society Award from IEEE Signal Processing Society (SPS). This is the highest award given by SPS for his work in multimedia security and image and video compression.



Narong Mettripun received a MSc degree in computer engineering from King Mongkut's University of Technology Thonburi (KMUTT), Thailand, in 2004. He is currently a lecturer in the Electrical Engineering Department, Rajamangala University of Technology Lanna Chiang Rai. He is now pursuing a PhD degree in electrical and computer engineering at KMUTT. From July 2011 to July 2012, he was a visiting researcher at Video and Image Processing Laboratory,