



Semi-fragile spatial watermarking based on local binary pattern operators

Zhang Wenyin ^{a,b,*}, Frank Y. Shih ^b

^a School of Informatics, Linyi University, Linyi, Shandong, 276005, China

^b College of Computing Sciences, New Jersey Institute of Technology, Newark, NJ 07102, USA

ARTICLE INFO

Article history:

Received 14 August 2010

Received in revised form 23 February 2011

Accepted 1 April 2011

Available online 15 April 2011

Keywords:

Digital watermarking

Multi-level watermark

Semi-fragile watermark

Local binary pattern

ABSTRACT

Local binary pattern (LBP) operators, which measure the local contrast within a pixel's neighborhood, have been successfully applied to texture analysis, visual inspection, and image retrieval. In the paper, we present a novel semi-fragile spatial watermarking method based on LBP operators by using the local pixel contrast for the embedding and extraction of watermarks. We also propose a general framework for multi-level image watermarking. Experimental results show that the proposed watermarking methods are robust against commonly-used image processing operations, such as additive noise, luminance change, contrast adjustment, color balance, and JPEG compression. At the same time, they achieve good invisibility, fragility, and image tamper detection and localization with less computational cost.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

The success of the Internet and digital consumer devices has been profoundly changing our society and daily lives by making the capture, transmission, and storage of digital data extremely easy and convenient. However, this raises a big concern on how to secure these data and prevent unauthorized modification. This issue has become problematic in many areas, such as copyright protection [1–4], content authentication [6,7], information hiding [5], and covered communications [8]. Many researchers have developed various algorithms of digital watermarking to address this issue [9–22], which intend to embed some secret data (called watermark) in digital content to mark or seal the digital data content. The watermark embedded into a host image is in such a way that the embedding-induced distortion is too small to be noticed. At the same time, the embedded watermark must be robust enough to withstand common degradations or deliberate attacks.

During last 20 years, digital watermarking techniques have achieved a big progress, from spatial domain to transformed domain, from robustness to fragility, and from irreversibility to reversibility. The earliest work of digital watermarking schemes can be traced back to the early 1990s, which presented the Least-Significant Bit (LSB) method to embed watermarks in the LSB of the pixels in spatial domain [5]. Patchwork methods [6,7] process pairs of pixels of the image to embed or extract watermarks. Spread-spectrum modulation

techniques [8] embed information by linearly combining the host image with a small pseudo noise signal that is modulated by the embedded watermark.

In the frequency domain, watermarks are inserted into the coefficients of a transformed image, for example, using the Discrete Fourier Transform (DFT) [9–12], Discrete Cosine Transform (DCT) [13–15], and Discrete Wavelet Transform (DWT) [16–18]. There are more literatures in frequency domain than in spatial domain, mainly because watermarking in the frequency domain can be easily combined with the human visual system (HVS). Recently, more attention is paid to reversible watermarking and tamper detection and recovery, such as [19–22].

In this paper, we introduce the local binary pattern (LBP) operators into image watermarking fields. The original LBP operator, which measures the local contrast of pixels, is widely used in the texture classification and face recognition [23–26]. By its extension, we define Boolean function operations on calculating LBP patterns, and adjust one or more of the pixels in the neighborhood to make the function results consistent with the bits of embedded watermarks to realize watermark embedding in spatial domain. Firstly, we explain the principle of watermark embedding and extraction processes by using the single-level watermarking technique. Then we discuss the technique of applying multilevel watermarking methods based on the LBP operators of different scale or radii and other improved LBP operators, such as Improved LBP [27] and Complete LBP [28].

The remainder of this paper is organized as follows. In Section 2, we introduce the basic knowledge of LBP operators. In Section 3, we propose the spatial single-level watermarking technique based on LBP operators. The experimental results and analysis are presented in Section 4. Section 5 provides a multilevel watermarking scheme and its analysis. Finally, we conclude the paper in Section 6.

* Corresponding author at: School of Informatics, Linyi University, Linyi, Shandong, 276005, China.

E-mail addresses: Zwy218@hotmail.com (Z. Wenyin), shih@njit.edu (F.Y. Shih).

2. Local binary pattern operator and its extensions

The local binary pattern (LBP) operator was proposed to measure the local contrast in texture analysis [22,23]. It has been successfully applied to visual inspection and image retrieval [24,25]. The LBP operator is defined in a circular local neighborhood. Using the center pixel as the threshold, its circularly symmetric P neighbors within a certain radii R are individually labeled as 1 when the value is larger than the center, or labeled as 0 when the value is smaller than the center. Note that $P = (2R+1)^2 - 1$. Then, the LBP code of the center pixel is produced by multiplying the thresholded values (i.e., 1 or 0) by weights given to the corresponding pixels, and summing up the result. For example, the LBP of a 3×3 window (where $R=1$ and $P=8$) uses the center pixel as a threshold value, and the values of the thresholded neighbors are multiplied by the binomial weight and summed to obtain the LBP number. In this way, the LBP can produce a number from 0 to 255. The entire LBP numbers composite a texture spectrum of an image with 256 gray levels, which is often used to extract image features for classification or recognition.

Given parameters P and R , which control the quantization of the angular space and spatial resolution respectively, the LBP number, denoted by LBP_p , indicating the local contrast in the neighborhood, is defined as

$$LBP_p = \sum_{p=0}^{P-1} S(g_p - g_c) \times 2^p \quad (1)$$

where g_c denotes the gray level of the center pixel c in the P neighborhood, g_p denotes the gray level of the neighboring pixels p , and $S(x)$ refers to the sign function defined as

$$S(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

More detailed information about the LBP operators and their applications can be referred to [25,26]. Zhang and Jin [27,29] improved the LBP operator by considering the magnitude of gray-level differences, concentrating on the visually most important texture pattern parts of images, and disregarding the unimportant details. They applied it on gas/liquid two-phase flow pattern analysis and recognition successfully. Guo et al. [28] presented another modeling of the local binary pattern operator for texture classification using two complementary components: the signs and the magnitudes.

3. The proposed spatial watermarking based on LBP operators

3.1. Definitions on a (P, R) local region

Before presenting the proposed watermarking algorithms, we first provide some definitions. Let g_c denote the gray level of the

center pixel c in the P neighborhood, and let g_p denote the gray level of the neighboring pixels p . For a (P, R) local region, we describe it as follows:

$$g_p = \{g_i | i = 0, \dots, c, \dots, P-1\} \quad (3)$$

$$m_p = \{m_i | m_i = |g_i - g_c|, i = 0, \dots, P-1\} \quad (4)$$

$$s_p = \{s_i | s_i = \text{sign}(g_i - g_c), i = 0, \dots, P-1\}. \quad (5)$$

Note that Eq. (5) uses the *sign* function, which is equivalent to Eq. (2). In this way, we divide the local region into three parts [28]: g_p is a vector composed of P pixels in the R radius, m_p is a vector built by the magnitude obtained from the difference between the p pixels and the center pixel g_c , and s_p is a sign vector from the difference. Fig. 1 shows an example of the three parts in a $(P=8, R=1)$ local region.

In order to embed watermarks, we define Boolean functions $f(s_p)$ to be applied on the binary sign vector part s_p . Two types of Boolean functions are chosen for illustration purposes, which are defined as follows:

$$f_{\oplus}(s_p) = s_0 \oplus s_1 \oplus \dots \oplus s_{P-1} \quad (6)$$

$$f_{\#}(s_p) = \text{Bool}\left(1(s_p) - 0(s_p) > N\right). \quad (7)$$

In Eq. (6), \oplus is the Exclusive OR (XOR) operator. Obviously, $f_{\oplus}(s_p) \in \{0, 1\}$. It satisfies the associative and commutative properties, so any circular bit shifted on s_p clockwise or counterclockwise does not change the function value. However, any one bit change in s_p from 0 to 1 or from 1 to 0 will reverse the function value.

In Eq. (7), $\#1(s_p)$ means the number of pixels with value “1” in s_p , $\#0(s_p)$ is the number of “0” in s_p , N is an integer, and $N \leq P-1$. If $\#1(s_p) - \#0(s_p) > N$, then $f_{\#}(s_p)$ returns 1; otherwise, it returns 0. In this way, $f_{\#}(s_p)$ is immune to bit shift and rotation.

3.2. Watermark embedding algorithm

We embed the watermarks by changing the value of $f(s_p)$ in a local region. The value of $f(s_p)$ is changed by altering the bits in s_p . These changes are reflected by modification of pixels in the spatial local region. Different Boolean functions correspond to different algorithms.

For instance, when using Boolean function $f_{\oplus}(s_p)$ in a (P, R) neighborhood, we select a pixel with the minimal magnitude in m_p to alter for embedding the watermark, so that the quality of the original image block will be affected the least. In other words, we keep the value of $f_{\oplus}(s_p)$ to be consistent with the corresponding bit of watermarks.

g_3	g_2	g_1	m_3	m_2	m_1	s_3	s_2	s_1
g_4	g_c	g_0	m_4		m_0	s_4		s_0
g_5	g_6	g_7	m_5	m_6	m_7	s_5	s_6	s_7
	g_8			m_8			s_8	

Fig. 1. An example of a $(8, 1)$ local region, as divided into three parts: g_8 is the pixel vector, m_8 is the magnitude vector, and s_8 is the sign vector.

The watermark embedding procedure can be summarized in the following two steps:

- 1) The original image is divided into (P, R) non-overlapping local region blocks. The LBP pattern is used to calculate m_p and s_p , as well as $f_{\oplus}(s_p)$. Let w be one of bits in the watermarks and β be the watermarking intensity factor.
- 2) For each (P, R) local neighborhood, if the value of $f_{\oplus}(s_p)$ equals to the value of w , we do nothing to the pixels in the neighborhood. Otherwise, we modify one of pixels by making the value of $f_{\oplus}(s_p)$ consistent with the corresponding w .

That is

if $(w = 1 \text{ and } f_{\oplus}(s_p) = 0) \text{ or } (w = 0 \text{ and } f_{\oplus}(s_p) = 1)$
then $\left\{ \begin{array}{l} \text{select } m_i = \min(m_p); \\ \text{if } s_i = 1 \text{ then } g_i = (g_i - m_i) \times (1 - \beta); \\ \text{else } g_i = (g_i + m_i) \times (1 + \beta) \end{array} \right\}$

■

Note that $\min()$ is the minimal function. If there are more than one minimum, we select anyone of the minimums to determine the pixel

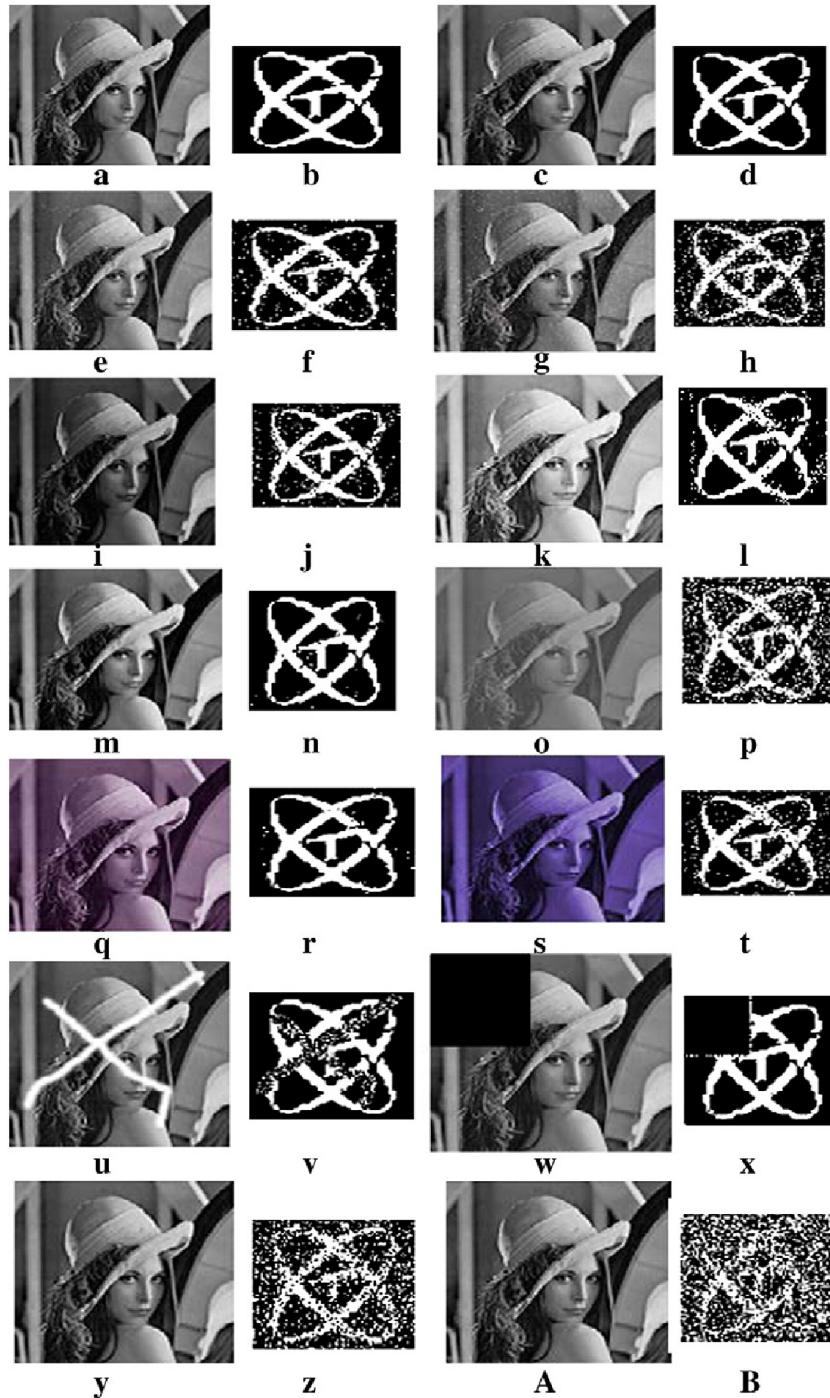


Fig. 2. Examples of applying some image-processing operations on the watermarked image. (a) The original Lena image, (b) the original watermark, (c) the watermarked Lena by the proposed algorithm with $\beta = 0.08$, (d) the extracted watermark with WNC = 1 and BNC = 1, (e) to the end: the results by applying different image-processing operations on (c). See context for more explanation.

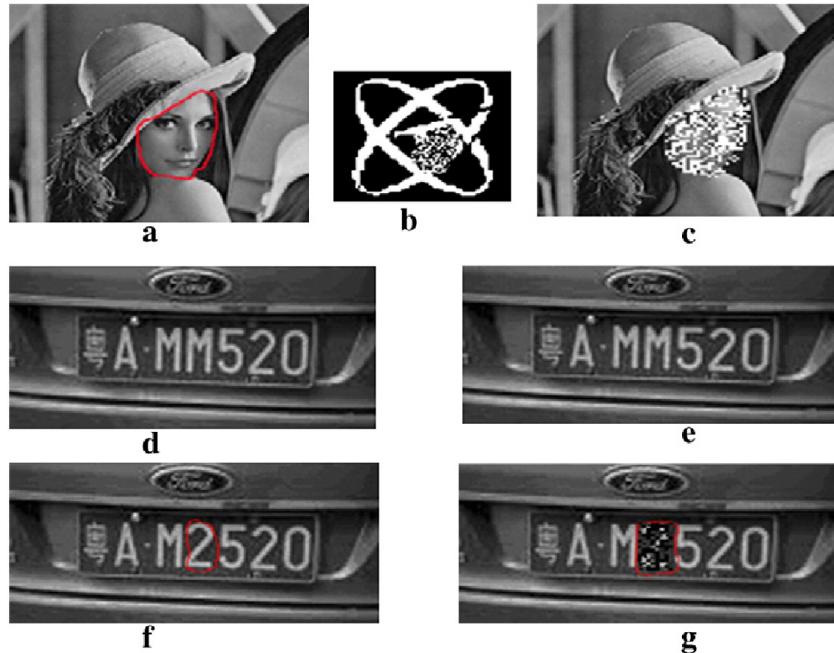


Fig. 3. Tamper detection and location. (a) The tampered image by replacing the face area, (b) the extracted watermark showing the tampering area, (c) tamper location, (d) original license plate, (e) watermarked license plate, (f) tampered license plate, (g) tamper detection and location.

to be changed. If a block's pixels are all “0” or “1”, we will modify the center pixel based on the corresponding watermarking bit before embedding it to the block.

3.3. Watermark extraction algorithm

The watermark extraction procedure in the proposed method becomes straight-forward. We judge the value of $f_{\oplus}(s_p)$ in the watermarked image to extract the watermark w . That is

$$\text{if } f_{\oplus}(s_p) = 1 \text{ then } w = 1 \text{ else } w = 0. \quad \blacksquare$$

4. Experimental results and analysis

We use the Lena image of size 256×256 to test the performance of the proposed algorithms. The watermark is a binary image of size 84×84 . The neighborhood is $(8, 1)$, which is a 3×3 local region. One local region embeds one bit of watermarks. Therefore, the watermarking capacity is $1/9$ of the original image size.

The notations are given below. $W(i, j)$ denotes the original watermark binary image of size $M \times M$, $W^*(i, j)$ denotes the extracted

watermarked binary image of size $M \times M$, $F(i, j)$ denotes the original image of size $N \times N$ to be watermarked, and $F^*(i, j)$ denotes the watermarked image. We use PSNR (peak signal-to-noise ratio), EBR (error bit rate), and NC (normalized correlation), as shown in Eqs. (8), (9), and (10), respectively, to evaluate the performance.

The EBR is used to compute the rate of error bits on the whole watermark accurate bits. The NC is used to locate a pattern on the extracted watermark image that best matches the specified reference pattern from the original image base [30]. Evidently, NC measures the amount of altered information which is originally “1”, and we name it as white NC (WNC). In order to accurately calculate the effect of the attack, the amount of altered information which is originally “0” is also considered, and we name it as black NC (BNC). Note that the formula of BNC is the same as Eq. (9) with all 1's being changed to 0's and vice versa. The PSNR is often used in engineering to measure the signal ratio between the maximum power and the power of corrupting noise. We use it to compare between the original and the embedded images in the spatial domain.

$$EBR = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} (W(i, j) \oplus W^*(i, j))}{M \times M} \quad (8)$$

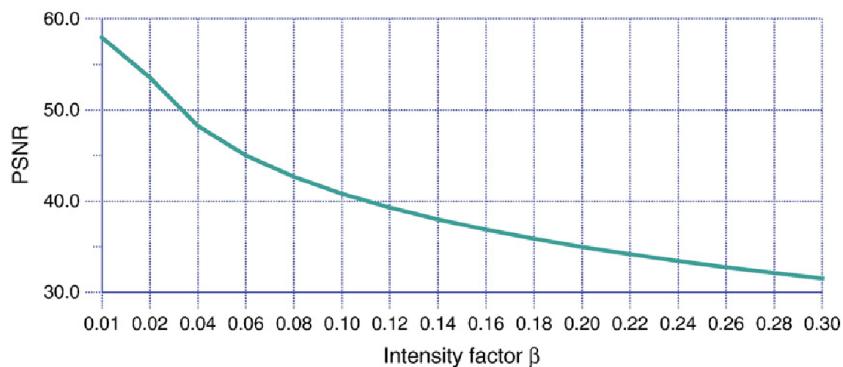


Fig. 4. The relationship between PSNR and intensity factor.

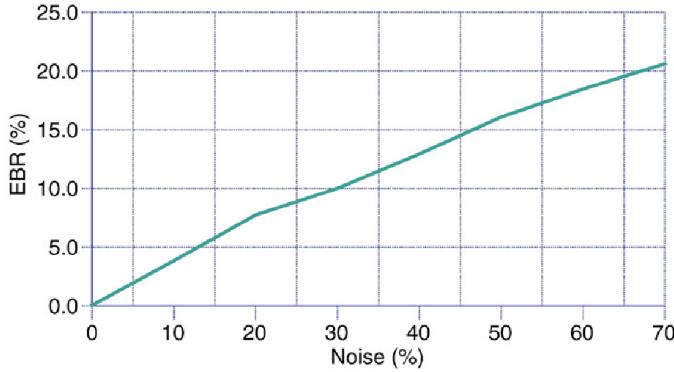


Fig. 5. The relationship between EBR and noise.

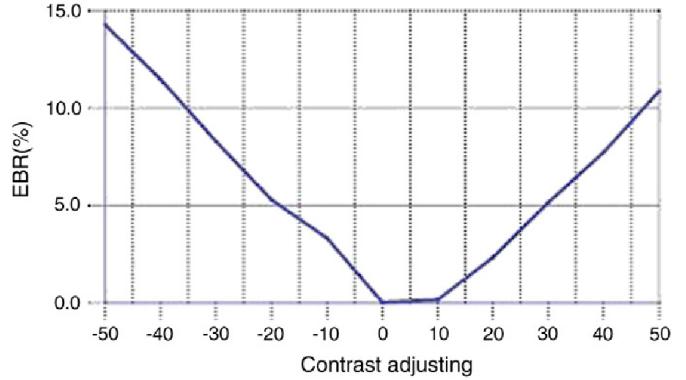


Fig. 7. The relationship between EBR and contrast adjustment.

$$NC = \frac{\sum_{i=1}^M \sum_{j=1}^M W(i,j)W^*(i,j)}{\sum_{i=1}^M \sum_{j=1}^M |W(i,j)|^2} \quad (9)$$

$$PSNR = 10 \log_{10} \left(\frac{255^2}{\sum_{i=1}^N \sum_{j=1}^N [F(i,j) - F^*(i,j)]^2 / N^2} \right). \quad (10)$$

By experiments, the proposed (8, 1) LBP based watermarking algorithm shows better transparency and robustness against some commonly-used image processing operations, such as additive noise, luminance variation, contrast adjustment, and color balance. Some examples of applying various operations on the watermarked image are shown in Fig. 2, where (a) is the original Lena image, (b) is the original watermark, (c) is the watermarked Lena by the proposed algorithm with PSNR 42.67 and intensity factor $\beta = 0.08$, and (d) is the extracted watermark with WNC = 1 and BNC = 1.

From Fig. 2(e) to the end, all processes are carried out on (c). Fig. 2(e) is the resulting image after adding 10% noise, and (f) is the extracted watermark with EBR = 3.85%, WNC = 0.959, and BNC = 0.962. Fig. 2(g) is the resulting image after adding 30% noise, and (h) is the extracted watermark with EBR = 10.01%, WNC = 0.887, and BNC = 0.905. Fig. 2(i) is the resulting image after logarithm transform of darkening, and (j) is the extracted watermark with EBR = 5.33%, WNC = 0.948, and BNC = 0.946. Fig. 2(k) is the resulting image after logarithm transform of brightening, and (l) is the extracted watermark with EBR = 2.98%, WNC = 0.979, and BNC = 0.966. Fig. 2(m) is the resulting image after contrast enhancement of +10%, and (n) is the extracted watermark with EBR = 0.47%,

WNC = 0.995, and BNC = 0.995. Fig. 2(o) is the resulting image after contrast reduction of -50%, and (p) is the extracted watermark with EBR = 14.24%, WNC = 0.875, and BNC = 0.851.

Fig. 2(q) is the resulting image after coloring by Photoshop 7.0, and (r) is the extracted watermark with EBR = 1.03%, WNC = 0.990, and BNC = 0.989. Fig. 2(s) is the resulting image after color saturation adjustment, and (t) is the extracted watermark with EBR = 5.75%, WNC = 0.946, and BNC = 0.940. Fig. 2(u) is the resulting image after destroying some parts, and (v) is the extracted watermark with EBR = 7.17%. Fig. 2(w) is the resulting image after cutting from the original image, and (x) is the extracted watermark. Fig. 2(y) is the resulting image after JPEG compression by Photoshop 7.0 with quality 12, and (z) is the extracted watermark with EBR = 19.20%, WNC = 0.80, and BNC = 0.81. Fig. 2(A) is the resulting image after JPEG compression with quality 11, and (B) is the extracted watermark with EBR = 35.12%, WNC = 0.648, and BNC = 0.656.

By experiments, the proposed method shows better image tamper detection ability. Fig. 3 provides two examples. In Fig. 3(a), the enclosed face area of the watermarked Lena image (see Fig. 2(c)) is replaced by the original (unwatermarked) face area in Fig. 2(a). The extracted watermark in Fig. 3(b) reveals the modification, and Fig. 3(c) shows the corresponding location of the modification.

Another example is the automobile license plate number forgery. Fig. 3(d) shows an original license plate image, and (e) is the watermarked image. Fig. 3(f) is the tampered image by using the digit "2" to replace the character "M" in the license plate of Fig. 3(e). Fig. 3(g) shows the result of tamper detection and location.

Figs. 4–8 show some validation results. From Fig. 4, as the intensity factor increases, the PSNR declines slowly, but maintains satisfactory values. When β reaches to 0.3, the PSNR is still above 30, which demonstrates that the proposed method keeps good image quality. Fig. 5

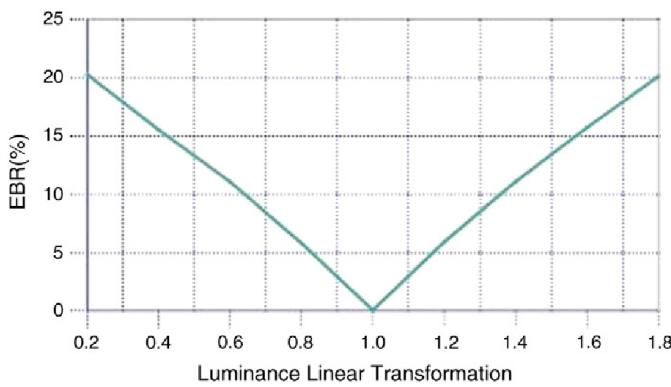


Fig. 6. The relationship between EBR and luminance linear transformation.

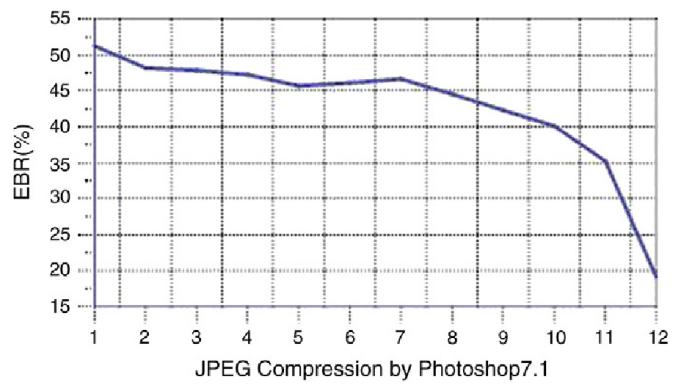


Fig. 8. The relationship between EBR and JPEG compression.

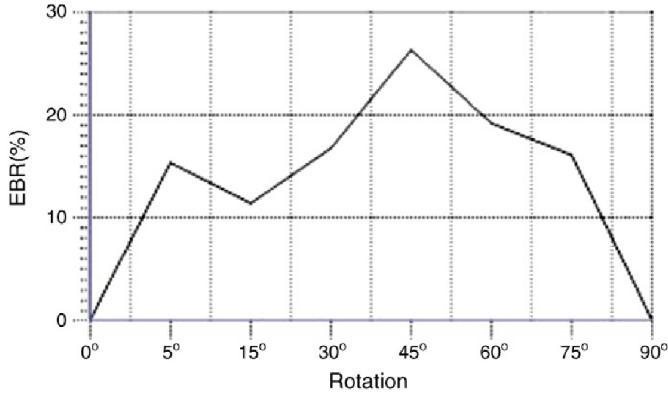


Fig. 9. The relationship between EBR and rotation.

shows its better power against noise. When the noise added is 50%, the watermarked image is nearly destroyed, but the EBR is only about 16%. It shows that the proposed method is very robust to noise. Fig. 6 embodies its good robustness against luminance modification. The best characteristic of the proposed method is its anti-contrast adjustment shown in Fig. 7, where the EBR keeps very low values, especially when contrast adjustment increases from 0 to 10. When contrast increases to 50% or decreases to -50%, the EBR are below 15%. Fig. 8 demonstrates that it is only robust against slight JPEG compression. When compression still keeps good quality, the method has EBR less than 20%. However, the proposed method is fragile to medium filter, image blurring, pixel interpolation, and other operations on a window neighborhood.

Although the function $f_{\oplus}(s_p)$ is invariant to rotation, the method achieves better results when the rotations are close to the multiples of 90°. It is because anyone of the bits in s_p changes from 0 to 1 or from 1 to 0, the value of $f_{\oplus}(s_p)$ will change into its inverse. To improve the robustness against rotation and compression, we use the function $f_{\#}(s_p)$ with $N=1$ and watermark intensity factor $\beta=0.02$. In experiment, we modify the center pixel to satisfy consistence between $f_{\#}(s_p)$ and the watermark bits. The watermark embedding algorithm is described as follows:

if $(w = 1 \text{ and } f(s_p) = 0)$ then do $\left\{ g_c = g_c \times (1 + \beta); \text{Compute } f(s_p); \right\} \text{while not } f(s_p)$ ■

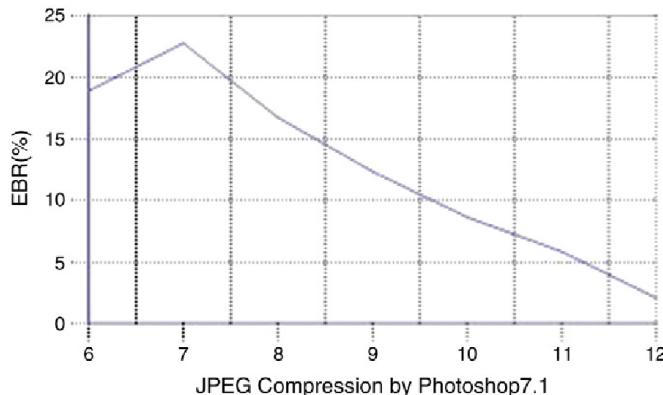


Fig. 10. The relationship between EBR and JPEG compression.

if $(w = 0 \text{ and } f_{\oplus}(s_p) = 1)$ then do $\left\{ g_c = g_c \times (1 - \beta); \text{Compute } f_{\#}(s_p); \right\} \text{while } f_{\#}(s_p)$.

By experiments, we observe that the function $f_{\#}(s_p)$ is more robust against additive noise, luminance change, contrast adjustment, JPEG compression, and rotation than the function $f_{\oplus}(s_p)$. Figs. 9 and 10 show the results with respect to rotation and JPEG compression. From Fig. 9, it is observed that when the watermarked image is rotated by 5°, 15°, 30°, 45°, 60°, 75°, and 90°, the EBR is 15.1%, 11.8%, 17.2%, 25.9%, 18.7%, 16.8%, and 0%, respectively. As the rotation angle is 45°, the result is the worst. Except 90°, the angle 15° corresponds to the best result. From Fig. 10, it is observed that when JPEG compression quality factor changes from 12 to 6 in the step of one, the EBR is 2.3%, 5.9%, 8.6%, 12.4%, 16.5%, 25.1%, and 18.2%, respectively. As the factor is 7, the EBR is the worst. With the decrease of the compression factors from 12 to 7, the EBR keeps an approximate linear increase.

5. Multilevel watermarking based on LBP operators

We can extend the aforementioned watermarking algorithm to multilevel watermarking techniques to achieve higher embedding capacity and better robustness. We firstly present a double-level watermarking algorithm and conduct analysis on its experimental results. Then, we extend it to a general framework for multilevel watermarking schemes.

5.1. Double-level watermarking

We divide the neighborhood s_p into two parts: even and odd neighbors, denoted as s_p^e and s_p^o . We perform $f_{\oplus}(s_p)$ on them and realize the embedding of two bits in the (P, R) neighborhood. In this way, the watermarking capacity is doubled. Fig. 11 shows an example of the $(8, 1)$ LBP pattern, which in fact is equivalent to two $(4, 1)$ neighborhoods.

An example of embedding two watermark images into the Lena image is shown in Fig. 12, where (a) is the original Lena image, (b) and (c) are two watermark images denoted by W1 and W2, and (d) is the watermarked image with PSNR = 36.5 and $\beta=0.08$. Fig. 12(e) and (f) are the two extracted watermarks from (d). Fig. 12(g) is the resulting image after adding 10% noise, and (h) and (i) are the extracted two watermarks with EBR = 1.96% and 2.47%, WNC = 0.980 and 0.966, BNC = 0.980 and 0.977, respectively. Fig. 12(j) is the resulting image after adding noise 120%, and (k) and (l) are the two extracted watermarks with EBR = 8.73% and 9.04%, WNC = 0.894 and 0.885, BNC = 0.919 and 0.916, respectively.

Fig. 12(m) is the resulting image after luminance reduction of -50%, and (n) and (o) are the two extracted watermarks with EBR = 10.23% and 7.74%, WNC = 0.916 and 0.949, BNC = 0.890 and 0.914, respectively. Fig. 12(p) is the resulting image after luminance enhancement of

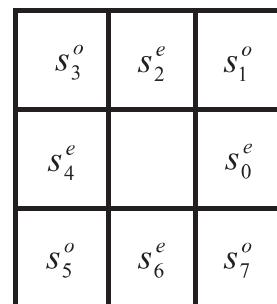


Fig. 11. The s_p^e and s_p^o of $(8, 1)$ LBP pattern. s_p^e denotes even neighbors and s_p^o denotes odd neighbors, $p=0 \dots 7$.

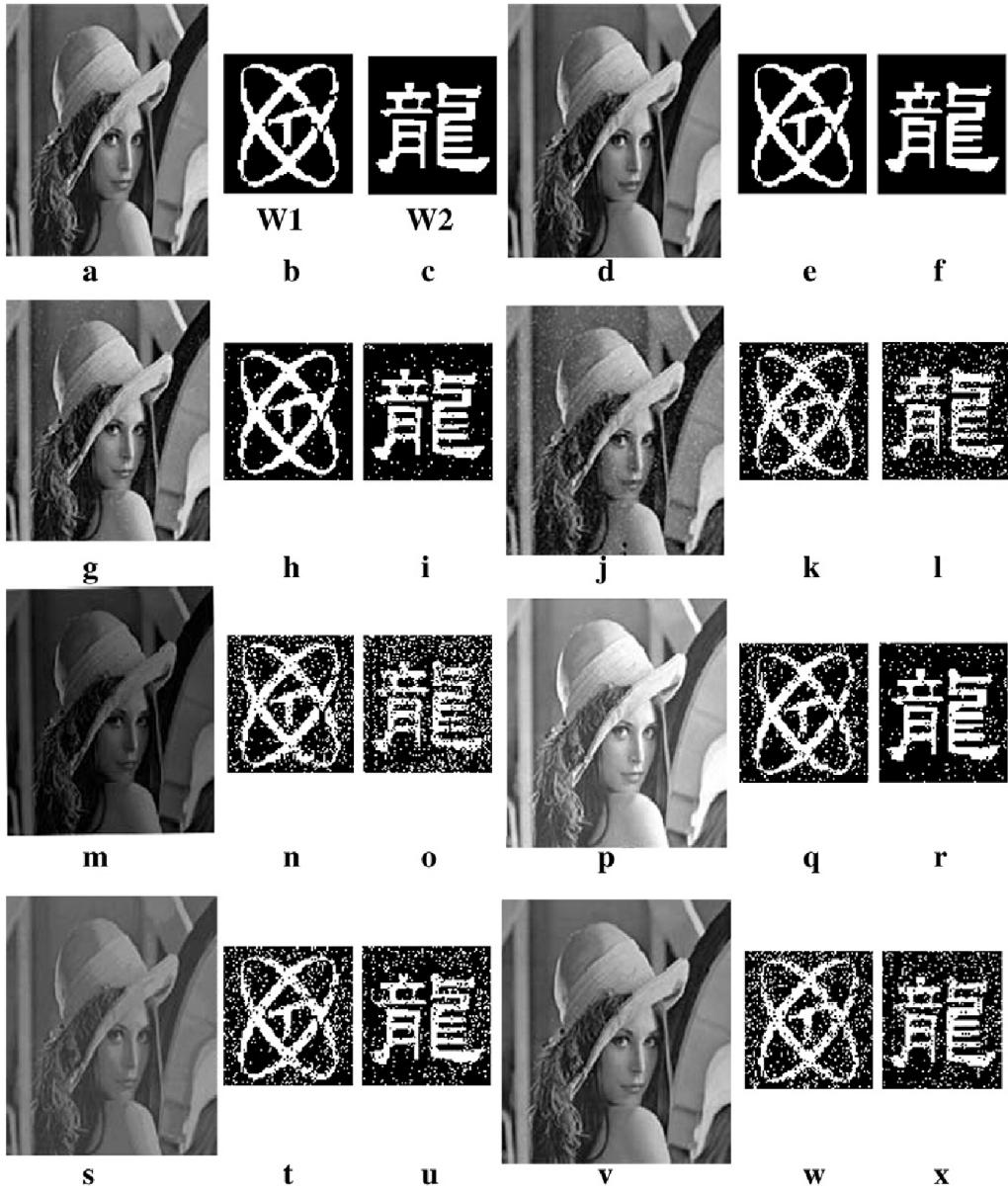


Fig. 12. Example of multilevel watermarking based on (8, 1) LBP pattern. (a): The original Lena image, (b) and (c): two watermarks W1 and W2, (d): the watermarked image, (e) and (f): the two extracted watermarks, (g)(j)(m)(p)(s)(v): the resulting images by applying different image-processing operations on (d), (h)(i)(k)(l)(n)(o)(q)(r)(t)(u)(w)(x): the extracted watermarks from (g)(j)(m)(p)(s)(v), respectively. See context for more explanation.

+50%, and (q) and (r) are the two extracted watermarks with EBR 10.08% and 7.55%, WNC = 0.923 and 0.946, BNC = 0.916 and 0.949, respectively. Fig. 12(s) is the resulting image after contrast reduction of

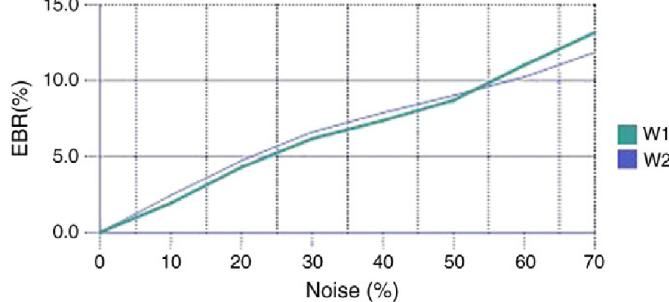


Fig. 13. The relationship between EBR and noise by double-level watermarking.

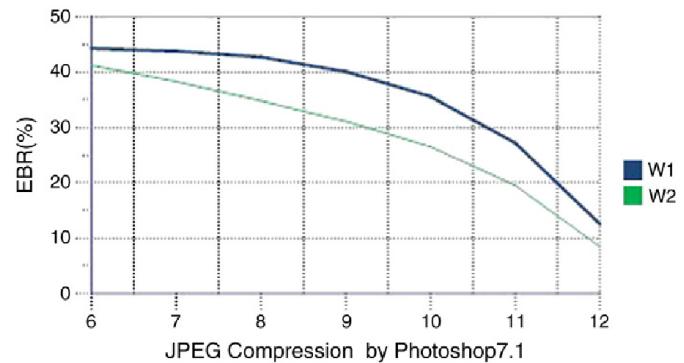


Fig. 14. The relationship between EBR and JPEG compression by double-level watermarking.

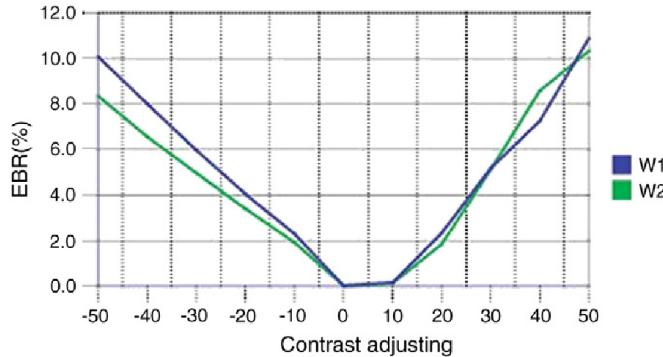


Fig. 15. The relationship between EBR and contrast adjustment by double-level watermarking.

–50%, and. (t) and (u) are the two extracted watermarks with EBR = 10.07% and 8.86%, WNC = 0.918 and 0.942, BNC = 0.892 and 0.908, respectively. Fig. 12(v) is the resulting image after JPEG compression with quality 12 by Photoshop 7.0, and (w) and (x) are the two extracted watermarks with EBR = 12.47% and 8.42%, WNC = 0.882 and 0.925, BNC = 0.872 and 0.912, respectively.

Note that the embedding and extraction of two watermarks do not interfere with each other. Figs. 13–16 show the performance curves after applying some image-processing operations. We observe that the double-level watermarking technique performs better robustness than the single-level one. In Fig. 13, when the double-level watermarked image is added by 50% noise, the two extracted watermarks are EBR 8.73% and 9.04%, but for single-level watermarking, EBR is 16.02%. In Fig. 14, when the double-level watermarked image is compressed by JPEG with quality factor 12, the two extracted watermarks are EBR 12.47% and 8.42%, but for single-level watermark, EBR is 19.02%. In Figs. 15 and 16, when the double-level watermarked image is applied by luminance or contrast adjustment, the two extracted watermarks are EBR 3%–5% lower than the single-level one.

5.2. Extension to multilevel watermarking

Based on double-level watermarking, we can extend it to multilevel watermarking using variant (P, R) blocks to embed multiple watermarks. For example, four-level watermarking on the 5×5 neighborhood block is shown in Fig. 17, which is divided into four parts: $s_i^1, s_i^2, s_j^3, s_j^4, i=0 \dots 3, j=0 \dots 7$. For s_i^1 and s_i^2 , we use $f_{\oplus}(s_p)$ to embed watermarks. For s_j^3 and s_j^4 , we use $f_{\#}(s_p)$ on anyone to embed watermarks and use $f_{\oplus}(s_p)$ on the other to embed watermarks.

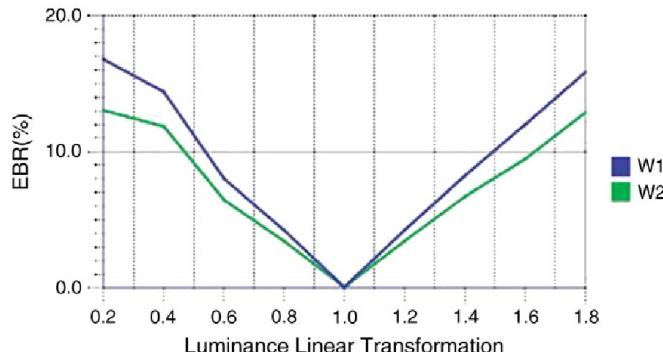


Fig. 16. The relationship between EBR and luminance linear transformation by double-level watermarking.

S_3^3	S_2^4	S_2^3	S_1^4	S_1^3
S_3^4	S_1^2	S_1^1	S_0^2	S_0^4
S_4^3	S_2^1		S_0^1	S_0^3
S_4^4	S_2^2	S_3^1	S_3^2	S_6^4
S_5^3	S_5^4	S_6^3	S_6^4	S_7^3

Fig. 17. The four parts of s_p in a 5×5 block. $s_i^1, s_i^2, s_j^3, s_j^4, i=0 \dots 3, j=0 \dots 7$ are used to embed the four watermarks, respectively.

Therefore, we can embed four watermarks individually without mutual interference.

Let $W_i, i=0 \dots 3$ be the four watermarks. In experiment, we firstly embed W_2 and W_3 , one of which is embedded by modifying the value of the center pixel (watermark factor $\beta=0.02$), and the other by changing one of non-center pixels (watermark factor $\beta=0.08$). Then, we embed W_0 and W_1 based on Section 5.1.

Fig. 18 shows some examples of multilevel watermarking. Fig. 18(a)–(d) are the original images of size 256×256 , and (e)–(h) are the four watermark images of size 51×51 . Fig. 18(i)–(l) are the watermarked images with PSNR 36.11, 35.01, 38.24, and 36.7, respectively, and the four watermark images can be extracted accurately. Because the embedding procedures of the four watermarks do not affect each other, their performances are basically consistent with the results provided previously in Sections 3 and 4.

Although the watermarked images achieve better PSNR, we can observe from Fig. 18 that some pixels in the smooth white or black region of these images are changed obviously, just like additive noises. In Fig. 18(l), we can see that several points are protruding in smooth regions, while in (j) it is difficult to see those points. Therefore, the proposed multilevel watermarking technique is very suited for the images with more complicated textures.

The proposed method can be similarly extended to other LBP operators with different (P, R) . We can design many multilevel watermarking schemes by jointly using $f_{\oplus}(s_p)$ and $f_{\#}(s_p)$ or using other different functions. Furthermore, the proposed method can be applied to the improved and complete LBP operators [27,28] to embed multilevel watermarks.

6. Conclusions

Based on the LBP operators, we propose a semi-fragile spatial watermarking scheme. The single-level and multilevel watermarking methods are described and analyzed. The proposed methods are robust against some commonly-used image processing operations, such as additive noise, luminance change, and contrast adjustment. At the same time, they maintain good fragility to some window operations, such as filtering and blurring, and have better sensitivity to image tampering. It can also achieve tamper detection and location.

For the future research, we will focus on the comprehensive comparison of different watermarking schemes based on different LBP

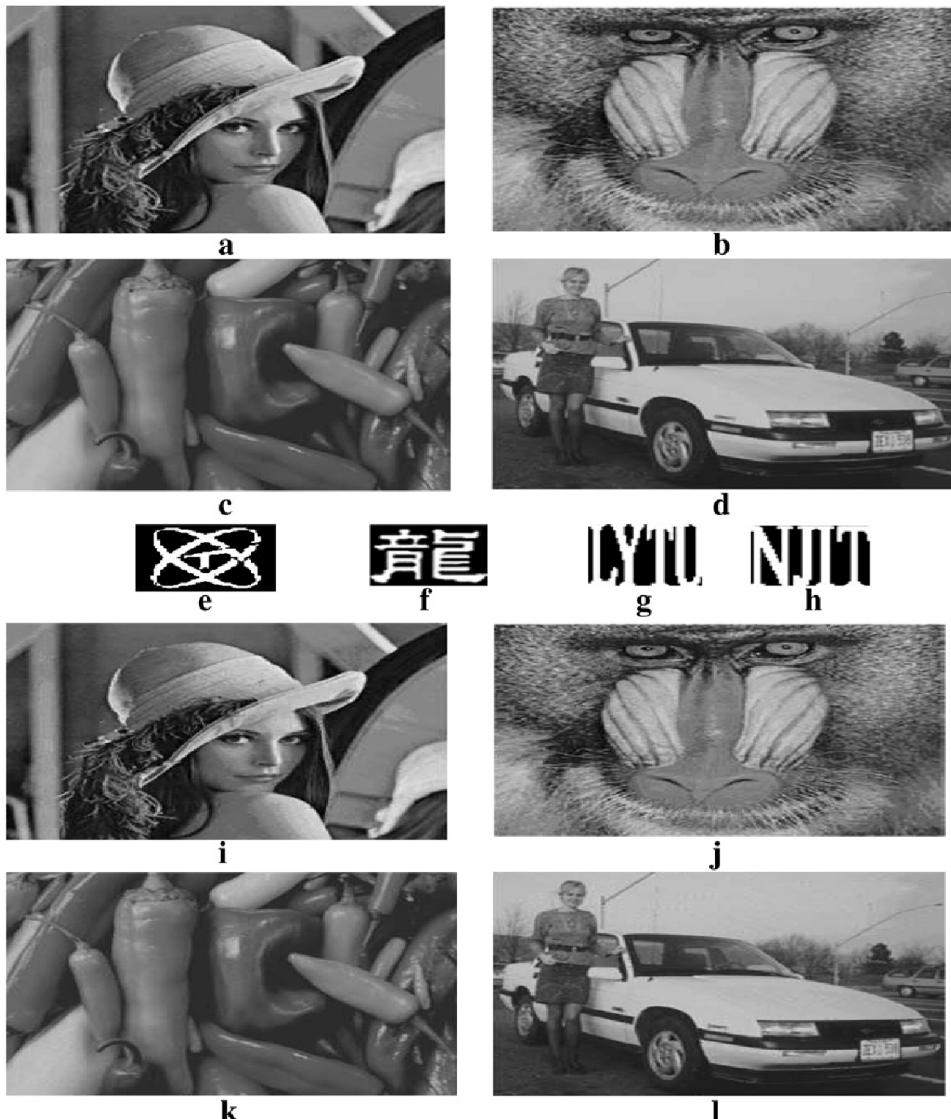


Fig. 18. Multilevel watermarking examples. (a)–(d) are the four original images, (e)–(h) are the four watermark images, and (i)–(l) are the watermarked images extracted from (e) to (h), respectively. See context for more explanation.

operators, their reversibility, and security. Also, we will conduct research on steganalysis based on LBP operators, as enlightened by [31].

References

- [1] M. Swanson, B. Zhu, A. Tewfik, Proc. IEEE Int. Conf. on Image Processing, vol. III, Sept. 1996, p. 211.
- [2] I. Pitas, Proc. IEEE Int. Conf. on Image Processing, vol. III, Sept. 1996, p. 215.
- [3] R. Schyndel, A. Tirkel, C. Osborne, Proc. IEEE Int. Conf. on Image Processing, vol. II, Nov. 1994, p. 86.
- [4] X. Xia, C. Bonelet, G. Arce, Proc. IEEE Int. Conf. on Image Processing, vol. I, Oct. 1997, p. 548.
- [5] K. Tanaka, Y. Nakamura, K. Matsui, Proc. IEEE ILCOM Int. Conf., 1990, p. 216.
- [6] I.-K. Yeo, H.-J. Kim, Proc. Int. Conf. Information Technology: Coding Computing, 2001, p. 237.
- [7] N. Cvejic, I. Tujkovic, Proc. IEEE Int. Symp. Consumer Electronics, U.K, 2004, p. 3.
- [8] B. Chen, G. Wornell, J. VLSI Signal Process, 2001, p. 7V33.
- [9] T.K. Tsui, X.P. Zhang, D. Androulacos, IEEE Trans. Forensics Security 3 (1) (March 2008) 16.
- [10] F.Y. Shih, Digital Watermarking and Steganography: Fundamentals and Techniques, CRC Press, Boca Raton, FL, 2008.
- [11] A. Reed, B. Hannigan, Proc. SPIE 4675, Apr. 2002, p. 222.
- [12] P. Bas, N.L. Bihan, J. Chassery, Proc. ICASSP, Hong Kong, China, Jun. 2003, p. 521.
- [13] I. Cox, J. Kilian, F. Leighton, T. Shamoon, IEEE Trans. Image Processing 6 (12) (Dec. 1997) 1673.
- [14] F.Y. Shih, S. Wu, Pattern Recognition 36 (2003) 969.
- [15] Z. Wei, K.N. Ngan, IEEE Trans. Circuits Systems Video Technology 19 (3) (March 2009) 337.
- [16] C. Podilchuk, W. Zeng, IEEE J. Sel. Areas Commun. 16 (4) (May 1998) 525.
- [17] N. Kaewkamnerd, K.R. Rao, Electron. Lett. 36 (2000) 518.
- [18] A. Piva, L. Boccardi, F.B.M. Barni, V. Cappellini, A.D. Rosa, Proc. IEEE Int. Conf. Multimedia Expo., New York, Jul. 30–Aug. 2, 2000, p. 1283.
- [19] L.X. Luo, Z.Y. Chen, M. Chen, X. Zeng, X. Zhang, IEEE Trans. Forensics Security 5 (1) (March 2010) 187.
- [20] H.-T. Wu, Y.-M. Cheung, IEEE Trans. Instrumentation Measurement 59 (1) (Jan. 2010) 221.
- [21] Y. Yang, X.M. Sun, H. Yang, C.T. Li, R. Xiao, IEEE Trans. Circuits Systems Video Technology 19 (5) (May 2009) 656.
- [22] V. Sachnev, H.J. Kim, J.N.S. Suresh, Y.Q. Shi, IEEE Trans. Circuits Systems Video Technology 19 (7) (July 2009) 989.
- [23] T. Maenpaa, M. Pietikainen, T. Ojala, Proc. 15th International Conference on Pattern Recognition, Barcelona, Spain, 2000, p. 951.
- [24] T. Ojala, M. Pietikainen, T. Maenpaa, IEEE Trans. Pattern Analysis Machine Intelligence 24 (7) (2002) 971.
- [25] T. Ahonen, A. Hadid, M. Pietikainen, Proceedings of European Conference on Computer Vision, Prague, Czech, 2004, p. 469.
- [26] T. Maenpaa, website, <http://herkules.oulu.fi/isbn9514270762/2003> Last accessed Dec. 20, 2008.
- [27] W.Y. Zhang, N.D. Jin, Proc. the 6th International Conference on Fuzzy Systems and Knowledge Discovery, August 2009.
- [28] G. Zhenhua, L. Zhang, D. Zhang, IEEE Trans. Image Processing 19 (6) (June 2010) 1657.
- [29] W. Zhang, F.Y. Shih, N. Jin, Y. Liu, Inter. J. Multiphase Flow 36 (2010) 793.
- [30] C.-M. Kung, S.-T. Chao, Y.-C. Tu, Y.-H. Yan, C.-H. Kung, J. Multimedia 4 (3) (June 2009) 112.
- [31] P. Lafferty, F. Ahmed, Proc. SPIE vol. 5561 (2004) 145.