Computers and Electrical Engineering 38 (2012) 1310–1324

Contents lists available at SciVerse ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

From single watermark to dual watermark: A new approach for image watermarking ☆

Hong Shen [a,b,*], Bo Chen [c]

[a] School of Computer and Information Technology, Beijing Jiaotong University, China
[b] School of Computer Science, University of Adelaide, Australia
[c] No. 38 Research Institute, China Electronics Technology Group Cooperation, China

ARTICLE INFO

Article history:
Available online 11 March 2012

ABSTRACT

Watermarking as a powerful technique for copyright protection, content verification, covert communication and so on, has been studied for years, and is drawing more and more attention recently. There are many situations in which embedding multiple watermarks in an image is desired. This paper proposes an effective approach to embed dual watermarks by extending the single watermarking algorithms in Xie and Shen (2005) [1] and Xie and Shen (2006) [2] for numerical and logo watermarking, respectively. Experimental results show that the resulting dual watermarking algorithms have a significantly higher PSNR than existing dual watermarking algorithms and also retain the same robustness as and higher sensitivity than the original single watermarking algorithms on which they are based.

1. Introduction

With the great development of digital media and increasing usage of Internet, digital watermarking [3] is gaining increasing popularity as an effective tool to protect ownership and data integrity of multimedia data [4]. Several image watermarking application scenarios were described in [5] and the desired properties of watermarks for each category were discussed. Different techniques have been developed to embed different types of watermarks into digital multimedia objects to achieve different goals. Basically, watermarks can be categorized into three types: the first type is intended to robustly carry ownership information to protect the copyright; the second type is intended to carry content-verification information; the third type is to convey side information [6]. Robust watermark, fragile watermark and covert communication are designed for these different application scenarios, respectively.

There are two broad types of watermarks for embedding: one is a numerical sequence, which is usually a pseudo-noise-like sequence and the embedding behaves like random noise [7]. Correlation calculation is the common technique used to detect this type of embedded watermarks. The other is a specific mark, which is usually a small meaningful logo. Logo watermarking has attracted more and more interest since the presentation of a recognizable logo is more perceivable than a numerical sequence.

Numerous image watermarking algorithms have been proposed in the last decade [8,9], and most of them embed watermarks in spatial domain or transform domain. Watermarks in spatial domain are easy to be destroyed, which are usually designed as fragile watermarks. Watermarks in transform domain distribute information all over, making them not easily perceivable. Robust watermarks are usually designed in transform domain. Because of the multi-resolution property of

☆ Reviews processed and approved for publication by Editor-in-Chief Dr. Manu Malek.
* Corresponding author at: School of Computer and Information Technology, Beijing Jiaotong University, China.
E-mail addresses: hongsh01@gmail.com (H. Shen), bochen@mail.ustc.edu.cn (B. Chen).

0045-7906/$ - see front matter © 2012 Published by Elsevier Ltd.
doi:10.1016/j.compeleceng.2011.11.012

the wavelet decomposition, more and more wavelet-based watermarking techniques considering human visual system (HVS) are proposed.

Traditional watermarking achieves only one goal. If we want to achieve several goals, for example, protect the copyright of one product with several owners in different phases, or verify the integrity of the content and protect the copyright at the same time, we have to turn to multiple watermarking, that is, embed more than one watermark into the same multimedia object. It has great advantages on multimedia data tracing, data usage monitoring, multiple property management. Most current multiple watermarking schemes use a double watermarking algorithm, which combines a robust watermark and a fragile watermark.

There are generally three schemes for dual watermark embedding as shown in Figs. 1–3. According to the features of the two watermarks to be embedded, we may embed them one after another or simultaneously. The first two schemes belong to the first type. In scheme 1, the fragile watermark is embedded into the robust watermark to generate a double watermark, which will be embedded into the original image to produce the final watermarked image. This scheme allows us to directly apply traditional watermarking techniques with minimal changes. There are some works [10,11] based on this scheme. However, in this scheme since the two watermarks are actually combined into one, attackers can break all protections by just breaking this embedded watermark. Another drawback of this scheme is that it does not lend itself to adding more watermarks after the embedding.

Scheme 2 works like a production line, watermarks can be embedded one after another dynamically. This is the advantage of this scheme, we can embed watermarks dynamically. An example of this scheme was given in [12]. The disadvantage of this scheme is that the embedding has more constraints than the first scheme, e.g., the latter embedding must have no or slight effect to the former embedding, making it harder to guarantee the quality of the watermarks.

Instead of embedding one after another, scheme 3 embeds two watermarks simultaneously as shown in Fig. 3. A dual watermarking system based on scheme 3 was proposed in [13], which embeds two watermarks in the DCT domain of the host image. This scheme can easily make two watermarks non-interfering with a proper division, thus provides an effective way to guarantee the quality of the two watermarks. With this scheme, traditional techniques can be adopted after some modification and we adopt to extend two existing single watermark embedding algorithms to dual watermarking under this approach. Some preliminary work of this paper was reported in [14].

Recent advances in watermarking research include watermarking using quantization index modulation (QIM) [15], transformation domain and evolutionary computation [16], and in an optimal feature region set [17], etc.

The rest of this paper is organized as follows. Section 2 shows some related work of different schemes. Sections 3 and 4 give two dual water marking algorithms using our approach. Section 5 provides some experimental results, Section 6 concludes the paper.

2. Related work

Chemak et al. [12] proposed a dual (double) watermarking algorithm based on scheme 2.

As depicted in Fig. 4, in their algorithm watermarks were embedded in both multi-resolution and spatial fields. The original image was first decomposed by 5/3 wavelet decomposition. Pixels with high intensity were selected from medium zone frequencies. Signature coding with Turbo code was embedded using insertion function $Y_i = X_i(1 + \alpha W_i)$, where *Yi* is the watermarked image, *Xi* is the original image, W_i is the mark to be embedded composed of 2000 bits coded with 1/2 ratio turbo coder, and α is the visibility coefficient set to 2 in their work. Watermarked image was rebuilt by inverse 5/3 wavelet decomposition. Then pixels with high luminosity were selected to do LSB2 substitution with the signature to get the final

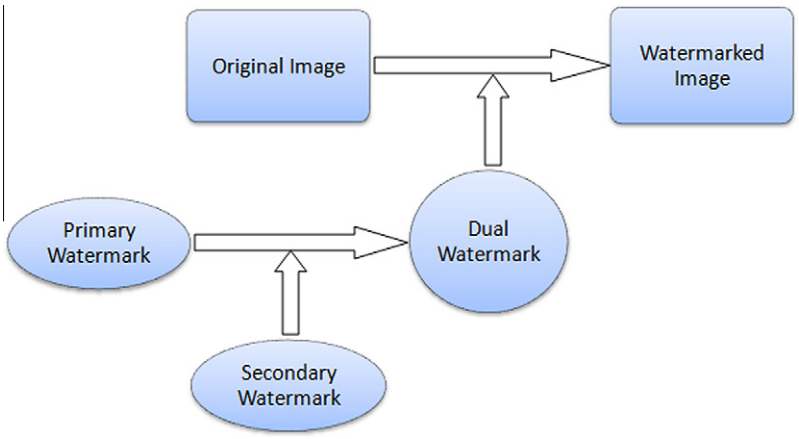


Fig. 1. Dual watermark embedding scheme 1.

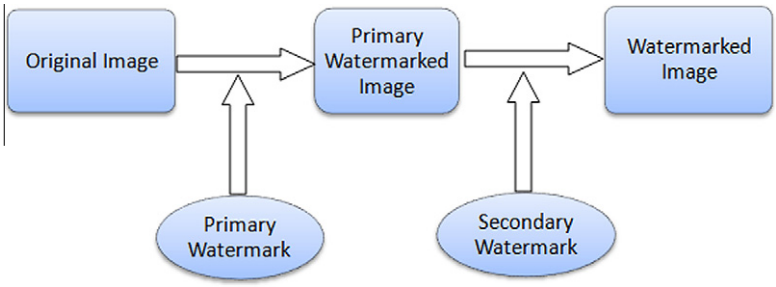


Fig. 2. Dual watermark embedding scheme 2.

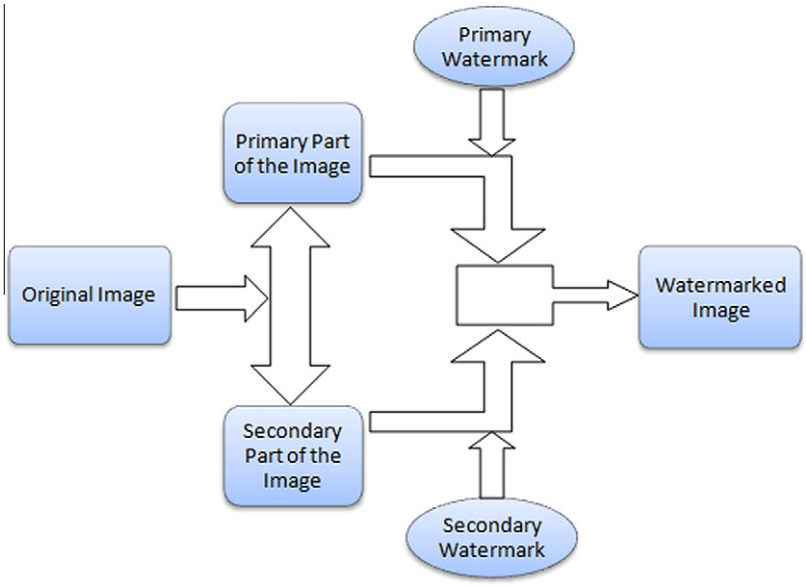


Fig. 3. Dual watermark embedding scheme 3.

dual watermarked image. Their main goal is to benefit from the advantages of watermarking process and to overcome the drawbacks of each individual watermark.

Habib et al. [13] embedded two watermarks in the DCT domain of the host image. Each DCT coefficient value was split into two parts, the integer part and the decimal part rounded to one decimal place. Each part formed an image with the same size as the host image. Then the robust watermark and the fragile watermark were embedded, respectively. Discarding the entire decimal part may result in some, though minor, data loss, which may cause quality degradation when returning to the spatial domain of the image. So retaining one decimal place in their system will make the watermarked image less lossy, and hence achieve better image quality. It also increases the image payload.

Osborne et al. [18] divided the medical image into two types of area: Region of Interest (ROI) and Region of Backgrounds (ROB). They developed a multiple watermarking technique to verify the integrity of the ROI after transmission.

First and Qi [19] proposed a blind grayscale logo watermarking system called CompMark to protect copyrights, which embeds a grayscale logo and its binary version in different wavelet-based multi-resolution sub-images using additive image fusion and modulus embedding. The two extracted logos are then combined to construct the final logo and a correlation value is used to determine the presence of the watermark. It gives a better performance than XFuseMark proposed in [2] and currently the best dual watermarking algorithm to our knowledge.

We will improve XFuseMark using our scheme which outperforms CompMark.

3. Our double watermarking algorithm

In this section, we first introduce two techniques on which our algorithm is based. One is the pixel-wise masking model proposed in [20] and improved in [1], the other is the pseudo-random sequence based bit substitution.

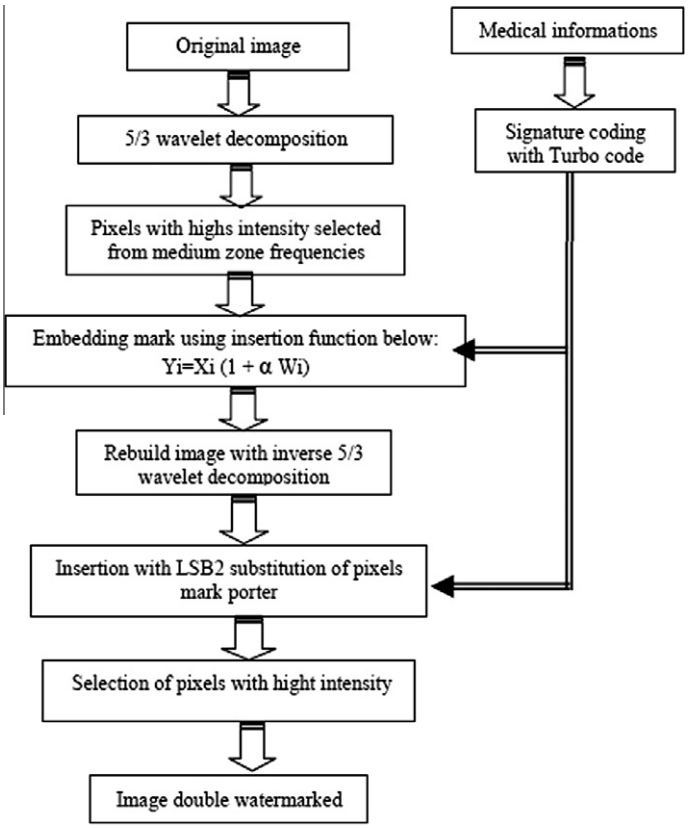


Fig. 4. Double watermark algorithm in [12].

3.1. Improved pixel-wise masking model

In [1], the watermark is embedded by modifying the wavelet coefficients using the rule

$$\widehat{I}_l^{\theta}(i,j) = I_l^{\theta}(i,j) + \alpha \cdot S_l^{\theta}(i,j) \cdot W^{\theta}(i,j) \cdot X^{\theta}(i,j). \tag{1}$$

Here, $I_l^{\theta}(i,j)$ denotes the subband at resolution level $l = 1, 2, \ldots, L$, and with orientation $\theta \in \{a,h,v,d\}$. X is the watermark, α is a global factor controlling the watermark energy. The pixel-wise masking matrix, W, is computed by a weighting function $W_l^{\theta}(i,j)$, which evaluates the local tolerance of the image to noise. The switching matrix, S, whose entry $S_l^{\theta}(i,j)$ determines whether the corresponding coefficient $I_l^{\theta}(i,j)$ is chosen to hold the watermark.

The weighing function $W_l^{\theta}(i,j)$ is computed as the product of three terms which are used to evaluate the sensitivity to noise changes depending on band, local brightness, and local texture activities, respectively.

$$W_l^{\theta}(i,j) = \Theta(l,\theta)\Lambda(l,i,j)\Xi(l,i,j)^{0.2}. \tag{2}$$

The orientation and resolution level are both considered in the first term, $\Theta(l,\theta)$, which is computed as follows:

$$\Theta(l,\theta) = \left\{ \begin{array}{cc} \sqrt{2}, & \text{if } \theta = d \\ 1, & \text{otherwise} \end{array} \right\} \cdot \left\{ \begin{array}{ll} 1.00, & \text{if } l = 1 \\ 0.32, & \text{if } l = 2 \\ 0.16, & \text{if } l = 3 \\ 0.10, & \text{if } l = 4 \\ 0.06, & \text{if } l = 5 \\ 0.03, & \text{if } l = 6 \\ 0.01, & \text{if } l \geqslant 7 \end{array} \right. . \tag{3}$$

The second term, $\Lambda(l,\theta)$, takes into account the local brightness of the reconstructed approximation subband at a given resolution, which is computed by the following formula:

$$\Lambda(l,i,j) = 1 + L'(l,i,j), \tag{4}$$

where

$$L'(l,i,j) = \begin{cases} 1 - I_l^a(i,j), & \text{if } I_l^a(i,j) < 0.5 \\ I_l^a(i,j), & \text{otherwise.} \end{cases} \tag{5}$$

This is based on the consideration that human eyes are less sensitive to changes in both very high and low brightness regions. $I_l^a(i,j)$, the entry of the reconstructed approximation subband, is normalized into the range [0, 1] before the computation in Eq. (5).

The third term, which measures the texture activity in the neighborhood of a coefficient, is computed as follows:

$$\Xi(l,i,j) = \frac{\left[I_l^h(i,j)\right]^2 + \left[I_l^v(i,j)\right]^2 + \left[I_l^d(i,j)\right]^2}{3} \cdot Var\{I_l^a(i,j)\}. \tag{6}$$

This term equals the product of two contributions: the first gives the local mean square value of the DWT coefficients at the corresponding resolution level in all detail subbands, which represents the distance from the edges; while the second, $Var\{I_l^a(i,j)\}$, is the local variance of the correspondingly reconstructed approximation subband in a small neighborhood (3×3 windows in [1]), which represents the texture activities. According to the consideration that human eyes are less sensitive in textured areas, but more sensitive near edges, the two terms should be multiplied [20].

The pixel-wise masking matrix for 512×512 Lena and 512×512 Girl image, are shown in Fig. 5. Entries are normalized into the range of [0, 1] for display.

3.2. Pseudo-random sequence based bit substitution

Bit substitution is a common technique in spatial watermarking algorithms, and LSB (Least Significant Bit) substitution is the most common one. However, LSB substitution cannot handle erase attacks. For example, with LSB2 adopted in [12] to embed the second watermark, if we remove the entire second significant bit plane and set it with all 1 or 0, every single bit embedded in LSB2 will be lost and hence the watermark will be totally removed. An improvement on basic LSB substitution using a pseudo-random number generator to determine the locations to be used for embedding was proposed in [21]. Still, the embedded watermark can be easily removed or altered by attackers once the locations are discovered.

To resist this kind of attacks, we propose to use another pseudo-random sequence based bit substitution. Its difference from other existing schemes is that we use pseudo-random sequence to determine the bits to be substituted rather than the locations. It is described as follows.

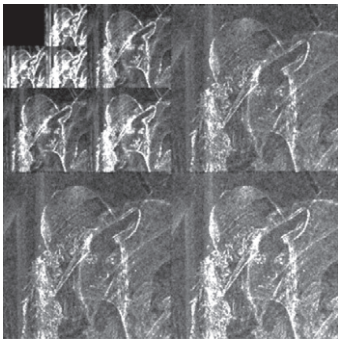
First, we generate a pseudo-random sequence with the given length (usually same as the length of the watermark) by a given "seed" which can be used as a secret key.

Defining an operator "bmod" which calculates the bit-wise modulus of the left operand over the right operand, we used a simple pseudo-random sequence generation function as follows:

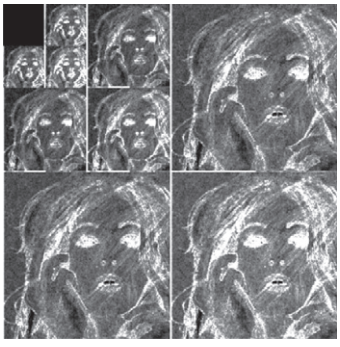
$$f(X,Y,Z) = X^{1/Y} bmod Z.$$

As a test case setting $X = 2$, $Y = 2$, $Z = 2$, we have,

$$\begin{aligned} f(2,2,2) &= 2^{1/2} bmod 2 \\ &= 1.4142134523\cdots bmod 2 \\ &= 1.0100111001\cdots \end{aligned}$$



(a) masking matrix for Lena



(b) masking matrix for girl

Fig. 5. The pixel-wise masking matrix.

In the above result there are 49,881 '0's and 50,119 '1's in $f(2,2,2)$'s 100,000 decimal bits. We use the decimal part of $f(2,2,2)$ as the index for substitution bit of every pixel. If the length of the watermark is excessively long, we can take a piece of the decimal part (e.g., 100 bits) and use it repeatedly to reduce the computation time, and add that with another parameter L denoting the decimal part length. The tuple (X, Y, Z, L) can be accessed as a secret key by embedder and extractor.

Here is a five-pixels example,

Before substitution we have

059	177	179	179	179
00111011	10110001	10110011	10110011	10110011

Substitution bits according to 01001 are shown below

0011101<u>1</u>	1011000<u>1</u>	1011001<u>1</u>	1011001<u>1</u>	1011001<u>1</u>

After substitution with watermark 01101, the result is

058	177	179	178	179
0011101<u>0</u>	1011001<u>1</u>	1011001<u>1</u>	1011001<u>0</u>	1011001<u>1</u>

In the above example, we have actually distributed the information into Z bit planes, which can resist erase attacks effectively: removal of the information (watermark) requires to remove at least Z bit planes, which will cause severe degradation in image quality and make the image unrecognizable in most cases.

3.3. Sensitivity evaluation

The switching matrix, $S_l^\theta(i,j)$ in Eq. (1), is very important for our algorithm. It helps us divide the wavelet coefficients into two parts, the insensitive (robust) part and the sensitive part. We apply the following procedure given in [1] to calculate the switching matrix:

- Denote the DWT of the host image and its compressed version by I and I^c, respectively. The compressed version can be produced by any compression tool such as JPEG. While compressing, use a small quality factor (10 in [1]) which can help us find those robust coefficients.
- Partition the detail subbands of I and I^c into blocks of a given size ($N = 16$ in [1]). Denote these blocks by two sets, $\{B_i | i = 1, 2, \ldots, N\}$ and $\{B_i^c | i = 1, 2, \ldots, N\}$, respectively.
- Compute the distortion of each pair of blocks by the mean square error function, i.e., $D_i = MSE(B_i, B_i^c)$. The less the distortion between B_i and B_i^c is, the more resilient to attacks the coefficients in B_i is. Therefore, D_i is a good evaluation for the attack sensitivity of the coefficients in the corresponding block.

Then, we sort the blocks, B_i, into a sequence by D_i from the lowest to the highest in which a certain (50 in [1]) percentage of the most leading blocks are chosen to embed watermarks. The entries of the switching matrix, $S_l^\theta(i,j)$, are set to one if their associated coefficients, $I_l^\theta(i,j)$, are chosen to do watermark embedding using Eq. (1), and these coefficients are called insensitive (robust) coefficients. The other entries of $S_l^\theta(i,j)$ are set to zero as the sensitive part.

3.4. Our double embedding scheme

Our algorithm generates two watermarks. The robust watermark of size N consists of pseudo-random binary entries which are generated from a private key. The fragile watermark of size N consists of all '0' or '1' or any other given binary sequence. Three levels DWT with Daubechies 9/7 filters is used in our algorithm. We set $\overline{S}_l^\theta(i,j) = 1 - S_l^\theta(i,j)$, another matrix for those sensitive coefficient, the fragile watermark is embedded there using our bit substitution scheme in three largest level-1 detail subbands. In order to do bit substitution, we need to cast those coefficients to integers, so we get integer version, $I_1'^\theta(i,j)$, of $I_1^\theta(i,j)$.

Then the two watermarks are embedded simultaneously by modifying the wavelet coefficients using the following rule.

$$\widehat{I}_l^\theta(i,j) = I_l^\theta(i,j) + \alpha \cdot S_l^\theta(i,j) \cdot W_l^\theta(i,j) \cdot X_l^\theta(i,j) + \overline{S}_l^\theta(i,j) \cdot bitset\left(I_l'^\theta(i,j), index, X_l'^\theta(i,j)\right). \tag{7}$$

When detecting the watermark, we first calculate the matrix $S_l^\theta(i,j)$ in the same way as used in embedding. Then we use the secret key (X, Y, Z, L) to get bit substitution index sequence with which the fragile watermark can be extracted and verified. After that, the robust watermark can be detected using the normalized correlation coefficient in the same way as in [1]:

$$\frac{\frac{1}{N}\sum_{i=1}^{N-1}[\widehat{I}(i)-\mu_1]\cdot[X(i)-\mu_2]}{\sigma_1\cdot\sigma_2}. \tag{8}$$

Here, $\widehat{I}$ and X denote the coefficients and the watermark, respectively, μ_1,μ_2 are the means of $\widehat{I}$ and X; σ_1, σ_2 are the standard deviations of $\widehat{I}$ and X; N is the size. The value of this equation is in range $[-1, +1]$. The greater it is, the more confident the detector is on the existence of the watermark.

4. Extension to a logo watermarking scheme

4.1. Logo embedding

The same idea of the above double watermarking can be applied to logo watermarking. The following steps are similar to the logo embedding procedure given by XFuseMark. The difference is we use XFuseMark as the first embedding in our double watermarking. Then, according to the content embedded we use an independent scheme to perform the second embedding which can be used to verify the copyright or to improve the robustness. Here we embed a binary version of the logo to enhance the extracted watermark and improve the robustness.

- *Step 1:* Get the Lth-level DWT wavelet coefficient matrix, f, of the host image using the Daubechies 9/7 filters, where $L \leqslant min(nr, nc)$. Because we need to make sure the size of all the subbands larger than or equal to the logo size.
- *Step 2:* Get the DCT coefficient matrix, $g(r, c)$, of the logo image. Eliminate some large-magnitude low-frequency components of g. The number of zeroed DCT coefficients at the top-left corner of the matrix g, recommended in XFuseMark is shown in Table 1. Since the large-magnitude DCT coefficients are set to zero, the remaining DCT coefficients are relatively small, the negative effects on the watermarked image quality by adding g into f are significantly reduced.
- *Step 3:* Partition the detail subbands of f into non-overlapping blocks of size $M_g \times N_g$, i.e., the size of the DCT coefficient matrix g. We denote these blocks by f_k, where k is the block index. XFuseMark uses a function, Q, to evaluate each blocks perceptual significance, all the partitioned blocks are ordered from most significant to least significant.

$$Q(k) = \frac{\sum_{r,c}[f^k(r,c)]^2}{M_g \cdot N_g}, \tag{9}$$

where $f_k(r,c)$ denotes the wavelet coefficients in the kth block. After the blocks are ordered by $Q(k)$ from most significant to least, the DCT coefficient matrix, g, is added repeatedly into the leading B percentage of these blocks. If we denote the used blocks by an index set J, the logo embedding process can be described by the following equation:

$$\hat{f}^k(r,c) = f^k(r,c) + \alpha \cdot R(g(r,c), \phi_k), \tag{10}$$

where $k \in J$, α is a global factor controlling the watermark strength, and R is a random permutation procedure depending on a private seed ϕ_k. The sequence of seeds ϕ_k for all the partitioned blocks in the wavelet domain, f, are generated from an initial secrete key. The value of B was set in [50, 80] in XFuseMark for an appropriate compromise between imperceptibility and robustness.

- *Step 4:* Obtain the binary version b of the logo image:

$$b(i,j) = \left\lceil \frac{g(i,j) - mean(g)}{max(g)} \right\rceil, \tag{11}$$

where $mean(g)$ is the mean of g which will used later. Then we get the binary sequence b' by doing b a Zig-Zag scan. And we embed b' into the rest of those blocks by our pseudo-random sequence based bit substitution.

- *Step 5:* Perform the inverse Lth-level DWT of the fused image components $\hat{f}^k(r,c)$ to get the final watermarked image $\widehat{I}(r,c)$.

The entire embedding process can be rewritten in the following abstract format:

$$\widehat{I}(r,c) = I(r,c) + \alpha \cdot W(r,c), \tag{12}$$

where I and $\widehat{I}$ denote the original host image and the watermarked image, respectively; a is a global factor controlling the watermark energy. W is the actually added signal generated by the above steps.

Table 1
XFuseMark's dimension configurations.

Host image	Logo image	Zeroed DCT coefficients
256×256	32×32	1×1
512×512	64×64	2×2
$\leqslant 1024 \times 1024$	128×128	4×4

4.2. Logo extraction

XFuseMark uses a weighted watermark estimation method to extract a distorted, recognizable version of the embedded logo. We do it in the same way, and then we extract the second watermark, the binary sequence, which is generated using the binary version of the logo image. The second watermark can help us enhance the logo image and thus improve the robustness. The extraction steps are as follows:

- *Step 1:* Decompose the watermarked image by the Lth-level inverse DWT. Denote the resulting wavelet coefficient matrix by $\hat{f}$. Partition the detail subbands into blocks of the size of the logo and order them from perceptually most significant to least as the embedder does. The leading B percentage of the blocks are used to extract the first watermark, the logo image. The rest blocks are used to extract the binary sequence.
- *Step 2:* Generate the same sequence of seeds, ϕ_k, from the initial key used in the embedder and inversely permute the blocks by $R^{-1}[\hat{f}^k, \phi_k]$, where $k \in \widehat{J}$. During this inverse permuting process, the DCT coefficients added into each block are moved back to their original positions.
- *Step 3:* Compute the reconstructed logo L in a weighted manner [2] as follows:

$$L(r,c) = IDCT\left(\sum_{k\in\widehat{J}} \frac{R^{-1}(\hat{f}^k(r,c),\phi_k)}{\widehat{Q}(k)}\right), \tag{13}$$

  where $\widehat{Q}(k)$ is the perceptual significance evaluation of the kth block f^k, $\widehat{J}$ approximates to the index set J used in the embedder.
- *Step 4:* Extract the binary sequence b' in the rest blocks according to the bit-substitution index and restore it to a binary logo b.
- *Step 5:* Adjust the normalized reconstructed logo, L, from its distorted version to the embedded logo. A nonlinear brightness adjustment given by the following equation is used in XFuseMark to enhance the brightness contrast of L.

$$\widetilde{L}(r,c) = \left(\frac{255}{255^{\lambda}}\right)\cdot L(r,c)^{\lambda}, \tag{14}$$

where λ is set to be 1.5, the transformation curve is showed in Fig. 6.

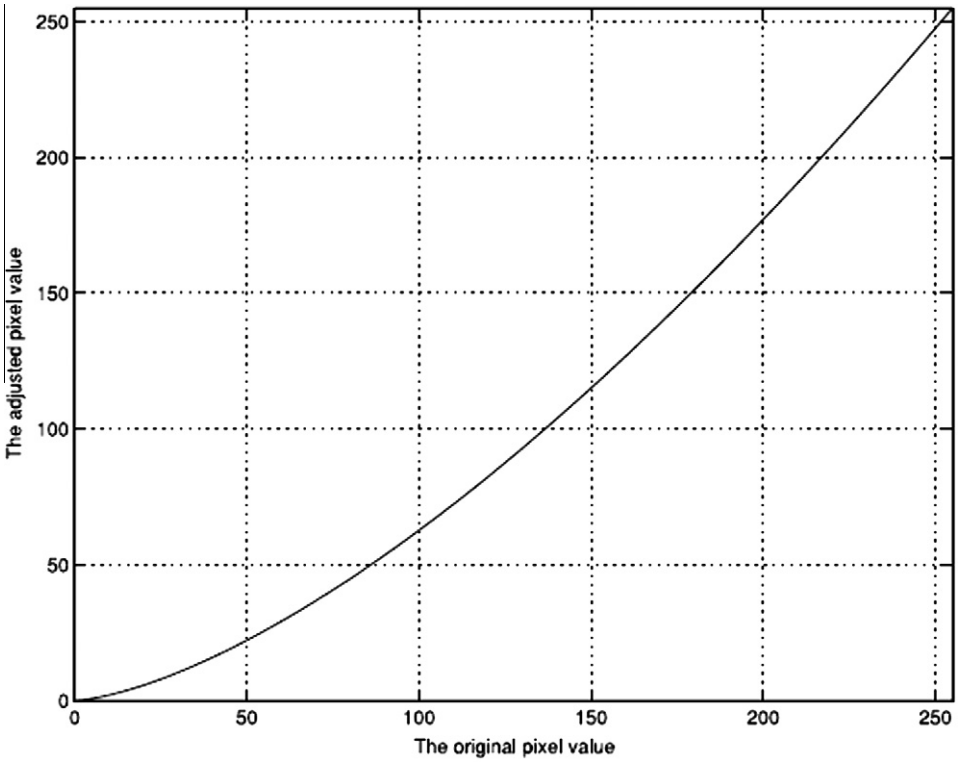


Fig. 6. The contrast enhancement function in XFuseMark.

(a) extracted logo (b) enhanced logo

Fig. 7. Comparison of the enhanced logo in XFuseMark and our scheme.

The drawback of this transformation is obvious: since the contrast enhancement function is fixed, the adjustment is blind and each different logo extracted will be adjusted in the same way. Since we get the binary logo b, we can involve b in the logo enhancement to make it more comfortable for normal viewers.

We use the following equation to adjust the logo image. Let m represent $mean(g)$.

$$\widetilde{L}(r,c) = \begin{cases} L(r,c) \cdot b(r,c) - m & \text{if } L(r,c) \geqslant m \\ L(r,c) + b(r,c) \cdot m & \text{if } L(r,c) < m \end{cases}. \tag{15}$$

And then adjust the mean value of the logo image in a linear way.

$$\widetilde{L} = \widetilde{L} - mean(\widetilde{L}) < m. \tag{16}$$

Fig. 7 shows the comparison of the enhanced logos in XFuseMark and our scheme. Obviously, our logo is much clearer and closer to the original logo.

4.3. Security analysis

The security of our scheme is based on the private key, the tuple (X, Y, Z, L) used in the bit substitution. The private key should be provided while extracting the secondary watermark embedded in those sensitive coefficients. If there is an active

(a) Original Image (b) Watermarked Image

Fig. 8. Lena image $\alpha = 0.5$, $PSNR_{lena} = 37.5868$.

(a) Original Image (b) Watermarked Image

Fig. 9. Girl image $\alpha = 0.5$, $PSNR_{lena} = 36.4830$.

(a) Original Image (b) Watermarked Image

Fig. 10. Pepper image $\alpha = 0.5$, $PSNR_{pepper} = 38.6069$.

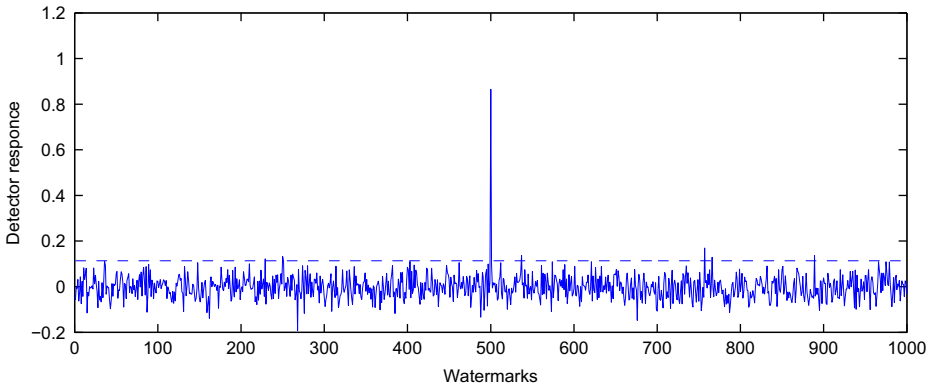


Fig. 11. The corresponding detector response to 1000 different watermarking codes.

attacker, he cannot get the exact substitution location without the key. And the brute force method is not working for such a large key space.

5. Experimental results

5.1. Numerical sequence watermarking

In the first set of experiments, we use a random sequence as primary watermark and a sequence of all '1's as the secondary watermark. We embed two watermarks into 512 × 512 Lena and 512 × 512 Girl images to test our double watermarking algorithm.

Figs. 8–10 show the watermarking results.

Fig. 11 gives the detector response of watermarked Lena image to different watermarking codes. When the detector detects the watermark, the correlation value will be a peak which is much bigger than other non-related watermarks.

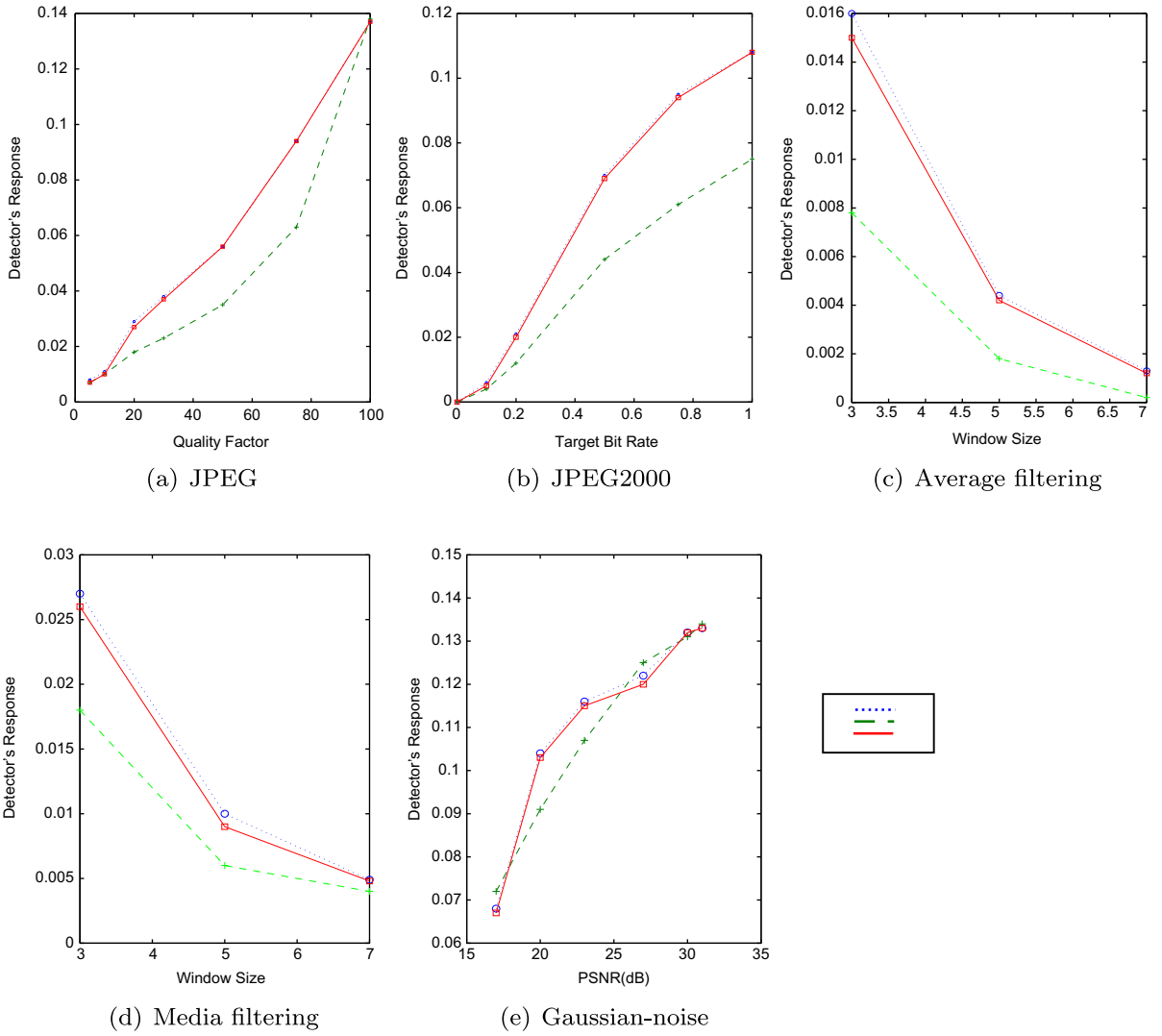
We compare [12,13] with our algorithm on the same set of test data. The results are shown in Tables 2 and 3.

Table 2
Correlation between the original watermark and the extracted robust watermark after attacks for the three models in [13] and our algorithm.

The attack	Model A	Model B	Model C	Our algorithm
Median filter	0.999	0.997	0.999	0.999
Rotate 90	0.999	0.992	0.999	0.999
Rotate 180	0.996	0.994	0.999	0.999
Crop and resize	0.996	0.994	0.999	0.999
Salt and pepper(2%)	0.973	0.970	0.970	0.980
Gaussian noise	0.960	0.960	0.960	0.975
JPEG-25	1	1	1	0.993
JPEG-50	0.998	0.994	0.994	0.995
JPEG-75	1	0.999	1	0.995

Table 3
PSNR comparison with Chemak's algorithm and our algorithm.

Distortion type	PSNR(Chemak's)	PSNR(Ours)
Mean shift	24.6090	28.5274
Contrast stretching	24.6003	28.5127
Impulsive salt and paper noise	24.6499	28.8416
Multiplicative speckle noise	24.6186	28.6741
Additive Gaussian noise	24.5906	28.4598
JPEG compression	24.7849	30.1737



(a) JPEG (b) JPEG2000 (c) Average filtering

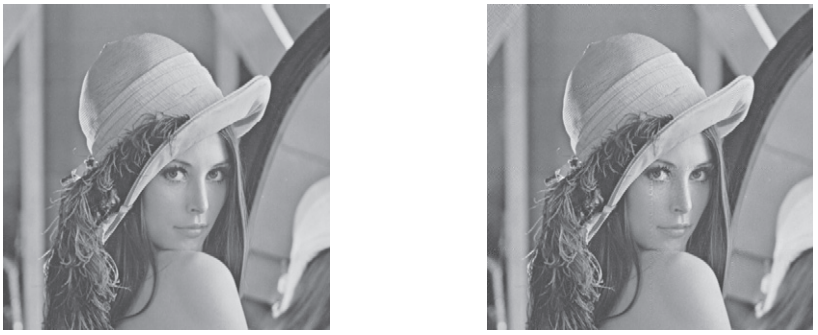
(d) Media filtering (e) Gaussian-noise

Fig. 12. The detector's response to the watermarked Lena image under different types of attacks. The dotted and dashed lines are Xie and Shen's algorithm and Barni's method, respectively, and the solid line is our algorithm.

Table 2 shows the correlation between the original watermark and the extracted robust watermark after attacks, from which we can see our algorithm performs better than Habib et al.'s algorithm in most of situations except JPEG compression attacks. This is because Habib et al.'s algorithm embeds watermarks in DCT domain which is compatible with JPEG compression while our work is done in DWT domain. But our algorithm still works well.

Besides, we calculate the PSNR value of watermarked image in Habib et al.'s algorithm which is not mentioned in [13], it is 30.1852 for Lena image and 30.1622 for Girl image. In algorithm, it is 40.0628 for Lena image and 39.4817 form Girl image with $\alpha = 0.1$. And when we let $\alpha = 0.5$, we still get 37.5868 for Lena image and 36.4830 for Girl image.

Fig. 13. Logo image used in the experiments.



(a) Original Image (b) Watermarked Image

Fig. 14. Watermarked Lena image $\alpha = 0.05$.

(a) Original Image (b) Watermarked Image

Fig. 15. Watermarked Girl image $\alpha = 0.05$.

In Table 3, we give a comparison of PSNR value between our algorithm and Chemak's algorithm after several types of distortions, from which, we can see that our algorithm achieves a better performance.

Fig. 12 shows the detector's response to the watermarked Lena image under several types of attacks, which is very similar to Xie and Shen's results and better than Barni's.

The experimental results show that our algorithm embeds two watermarks to the host image without introducing much noise, and the embedding has desirable properties – the robust watermark preserves a high robustness against attacks including filtering, noise addition and compression at the same level as [1] and the fragile watermark gives a high sensitivity against these attacks.

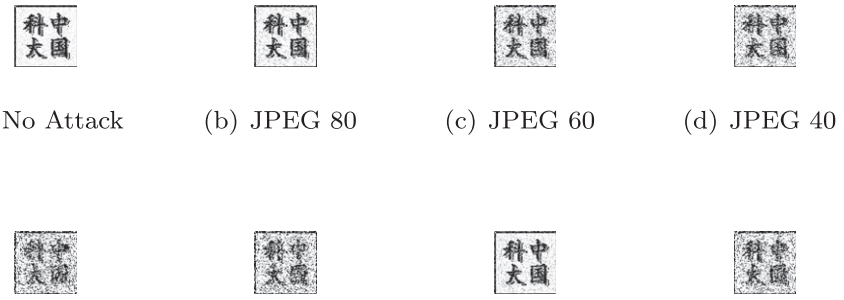
5.2. Logo watermarking

Fig. 13 is the 64×64 logo image used in our logo watermarking experiments.

Figs. 14–16 show the original and watermarked Lena, Girl and Peppers images. It is showed that, after the logo embedding with $\alpha = 0.05$, there is almost no perceptual changes in the image. There may be some visible artifacts in the watermarked images because for a detectable watermark it is theoretically impossible to make its embedding effect invisible.

(a) Original Image (b) Watermarked Image

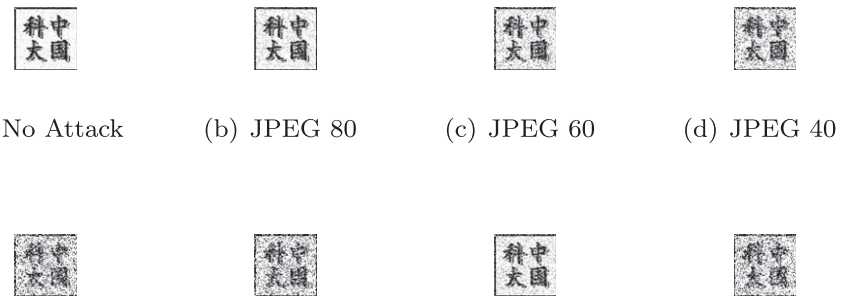
Fig. 16. Watermarked Peppers image $\alpha = 0.05$.



(a) No Attack (b) JPEG 80 (c) JPEG 60 (d) JPEG 40

(e) Mean 3×3 (f) Median 3×3 (g) Noise 4% (h) Noise 10%

Fig. 17. Logos extracted from watermarked Lena image under different image processing attacks.



(a) No Attack (b) JPEG 80 (c) JPEG 60 (d) JPEG 40

(e) Mean 3×3 (f) Median 3×3 (g) Noise 4% (h) Noise 10%

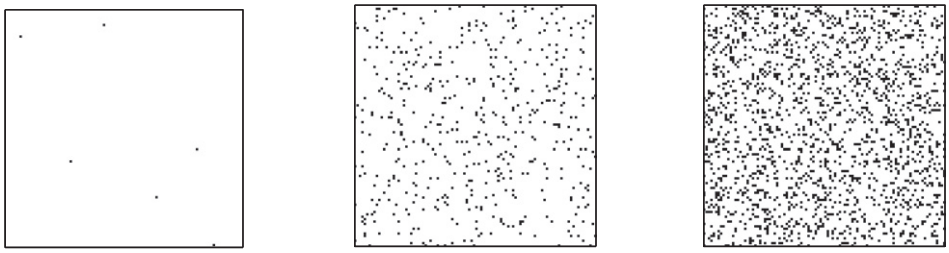
Fig. 18. Logos extracted from watermarked Girl image under different image processing attacks.

Table 4
The correlation value between the extracted and original Lena images.

Attacks	Value
No attack	0.9817
JPEG compression (quality = 80)	0.9351
JPEG compression (quality = 60)	0.7977
JPEG compression (quality = 40)	0.7188
Mean 3×3	0.5788
Median 3×3	0.6286
Noise 4%	0.9323
Noise 10%	0.6537

Table 5
The correlation value between the extracted and original Girl images.

Attacks	Value
No attack	0.9811
JPEG compression (quality = 80)	0.9323
JPEG compression (quality = 60)	0.8253
JPEG compression (quality = 40)	0.7047
Mean 3×3	0.5815
Median 3×3	0.6247
Noise 4%	0.8815
Noise 10%	0.6062



(a) no attack (b) Additive Gaussian noise (c) JPEG Compression

Fig. 19. Difference map of fragile watermark after attacks.

Figs. 17 and 18 are the experimental results showing the logos extracted from images watermarked by our scheme under different image processing attacks. As can be clearly seen in the figures, the extracted logos are easily recognizable, so our scheme is robust against a wide variety of image processing attacks.

The correlation values between the extracted logo and the original logo image are showed in Tables 4 and 5.

Compared with CompMark in [19], our extracted logos are more correlative with the original logo under the same attack.

5.3. Fragile watermark

Attacks are different for robust watermark and fragile watermark. Attackers are trying to destroy or alter the watermark to make detection failure in robust watermark. And in fragile watermark, attackers are trying to modify the content and avoid touching the watermark. Usually, there are two types of attacks for fragile watermark, intentional and unintentional. Transmission errors are unintentional attacks. These attacks are unavoidable, and little effect to the image. Those intentional attackers are trying to modify the content of the image with no watermark interfering.

Let $W'_f(i,j)$ be the extracted fragile watermark. $W_f(i,j)$ is the original one. $D(i,j)$ is the difference map.

$$D(i,j) = \left| W_f(i,j) - W'_f(i,j) \right|. \tag{17}$$

The effect from unintentional attack is like Gaussian distribution, which makes some isolated points in the difference map. But the intentional attack usually makes the extracted watermark totally different from the original one. The difference map $D(i,j)$ after some attacks is shown below. (The fragile watermark sequence is reshaped to rectangle for display convenience.) From Fig. 19 we can see our fragile watermark achieves a high sensitivity against attacks.

6. Conclusion

In this paper, we proposed a new approach for image watermarking. Following this approach, we developed a robust–fragile dual watermarking algorithm based on Xie and Shen's improved pixel-wise masking model [1] in combination with a novel pseudo-random sequence based bit substitution technique. In our algorithm, after adding a robust watermark into the most insensitive coefficients in the DWT of the image, those sensitive coefficients, which are not used in Xie and Shen's algorithm, are used to carry the fragile watermark. This makes the two watermarks non-interfering and increases watermarking capacity of the host image without reducing the watermark robustness. Following the same approach, another algorithm based on XFuseMark is also proposed to embed a meaningful logo and its binary version, which is more practical and robust for applications. It gives a better performance than CompMark in [19]. Our new approach provides an effective way to

transform a watermarking algorithm to transform from embedding a single watermark to dual watermarks while retaining the same robustness.

References

[1] Xie G, Shen H. Toward improved wavelet-based watermarking using the pixel-wise masking model. In: IEEE international conference on image processing (ICIP'05), vol. 1, 2005.
[2] Xie G, Shen H. A new fusion based blind logo-watermarking algorithm. IEICE Trans Inf Syst 2006:1173–80.
[3] Cox I, Miller M, Bloom J. Digital watermarking: principles and practice. Morgan Kaufman; 2001.
[4] Podilchuk C, Delp E. Digital watermarking: algorithms and applications. IEEE Signal Proc Mag 2001;18:33–46.
[5] Mintzer F, Braudaway G, Yeung M, Center I, Heights Y. Effective and ineffective digital watermarks. In: Proceedings of the international conference on image processing, vol. 3, 1997.
[6] Mintzer F, Braudaway G, Center I, Heights Y. If one watermark is good, are more better? In: Proceedings of the IEEE international conference on acoustics, speech, and signal processing (ICASSP'99), vol. 4, 1999.
[7] Cox I, Kilian J, Leighton F, Shamoon T. Secure spread spectrum watermarking for multimedia. IEEE Trans Image Process 1997;6:1673–87.
[8] Voyatzis G, Pitas I. Digital image watermarking using mixing systems. Comput Graph 1998;22:405–16.
[9] Kundur D, Hatzinakos D. Toward robust logo watermarking using multiresolution image fusion principles. IEEE Trans Multimedia 2004;6:185–98.
[10] Sharkas M, ElShafie D, Hamdy N. A dual digital-image watermarking technique. In: Proceedings of the third world enformatika conference, April.
[11] Wu K, Yan W, Du J. A robust dual digital-image watermarking technique. In: International conference on computational intelligence and security workshops (CISW'07), 2007. p. 668–71.
[12] Chemak C, Bouhlel M, Lapayre J. A new scheme of robust image watermarking: the double watermarking algorithm. In: Proceedings of the 2007 summer computer simulation conference, Society for Computer Simulation International San Diego, CA, USA, 2007. p. 1201–8.
[13] Habib M, Sarhan S, Rajab L. A robust–fragile dual watermarking system in the DCT domain. Lect Notes Comput Sci 2005;3682:548.
[14] Chen B, Shen H. A new robust–fragile double image watermarking algorithm. In: Proceedings of the third international conference on multimedia and ubiquitous engineering. p. 153–7.
[15] Phadikar A, Maity SP, Verma B. Region based QIM digital watermarking scheme for image database in DCT domain. Comput Electr Eng 2011;37:339–55.
[16] Lai C-C, Chen H-C, Yeh G-M, Ouyang C-S. A robust digital image watermarking using transformation domain and evolutionary computation techniques. In: ICMLC. p. 1643–7.
[17] Tsai J-S, Huang W-B, Kuo Y-H. On the selection of optimal feature region set for robust digital image watermarking. IEEE Trans Image Process 2011;20:735–43.
[18] Osborne D, Abbott D, Sorell M, Rogers D. Multiple embedding using robust watermarks for wireless medical images. In: Proceedings of the third international conference on mobile and ubiquitous multimedia, ACM, New York, NY, USA; 2004. p. 245–50.
[19] First E, Qi X. A composite approach for blind grayscale logo watermarking. In: International Conference Image Processing (ICIP 2007), 2007. p. III: 265–8.
[20] Barni M, Bartolini F, Piva A. Improved wavelet-based watermarking through pixel-wise masking. IEEE Trans Image Process 2001;10:783–91.
[21] Johnson N, Katzenbeisser S. A survey of steganographic techniques. In: Information hiding. p. 43–78.

Hong Shen is currently national "Thousand Talents" Professor in Beijing Jiaotong University, China, and tenured Professor of Computer Science in University of Adelaide, Australia. Received his Ph.D. from Abo Akademi University, Finland, he was Professor and Chair of Computer Networks Laboratory in Japan Advanced Institute of Science and Technology (JAIST) during 2001–2006, and Professor of Computer Science at Griffith University before that. He published more than 280 papers including over 100 papers in international journals such as a variety of IEEE and ACM transactions. He received many honors/awards, served on editorial boards of numerous journals and chaired several conferences.

Bo Chen received the B.Eng. degree in computer science and Ph.D. degree in computer software and theory from the University of Science and Technology of China in 2006 and 2011, respectively. Currently, he is a Researcher in No. 38 Research Institute, China Electronics Technology Group Cooperation.