A Novel Blind Multiple Watermarking Technique for Images

Peter H. W. Wong, *Member, IEEE*, Oscar C. Au, *Senior Member, IEEE*, and Y. M. Yeung

***Abstract*—Three novel blind watermarking techniques are proposed to embed watermarks into digital images for different purposes. The watermarks are designed to be decoded or detected without the original images. The first one, called single watermark embedding (SWE), is used to embed a watermark bit sequence into digital images using two secret keys. The second technique, called multiple watermark embedding (MWE), extends SWE to embed multiple watermarks simultaneously in the same watermark space while minimizing the watermark (distortion) energy. The third technique, called iterative watermark embedding (IWE), embeds watermarks into JPEG-compressed images. The iterative approach of IWE can prevent the potential removal of a watermark in the JPEG recompression process. Experimental results show that embedded watermarks using the proposed techniques can give good image quality and are robust in varying degree to JPEG compression, low-pass filtering, noise contamination, and print-and-scan.**

***Index Terms*—Blind watermarking, data hiding, multiple watermarks.**

I. Introduction

In RECENT years, watermarking has been an exciting topic and there have been many watermarking schemes proposed. Among these schemes, those requiring both the original data and the secret keys for the watermark bit decoding are called private watermark schemes. Those requiring the secret keys but not the original data are called public or blind watermark schemes [2]. Those requiring the secret keys and the watermark bit sequence are called semi-private or semi-blind watermark schemes [3]. Usually, the robustness of private watermark schemes is good under signal processing procedures such as JPEG compression and filtering. However, private schemes are not feasible in situations such as watermark detection in DVD players, because the original data is not available. Blind watermark schemes, on the other hand, detect the watermarks without the original data and are feasible in those situations. The trade-off is that the blind schemes are usually less robust and have relatively higher false alarm rate compared with the private schemes. This paper is about blind watermarking schemes.

There are many existing private schemes for robust watermarking. Cox *et al.* [1] uses spread spectrum to embed watermark in the discrete cosine transform (DCT) domain. To improve Cox's method, Lu *et al.* [4] uses cocktail watermark to improve the robustness and used human visual system (HVS) to maintain high fidelity of the watermarked image. Hsu *et al.* [5], [6] embeds watermark bits by modifying the polarity of DCT and discrete wavelet transform (DWT) coefficients and uses a meaningful logo image as the watermark. Huang *et al.* [7] embeds a watermark pattern by modifying the DC components.

There are also many blind watermark schemes. Hartung [8] assumes small correlation between the secret key and the image and hides data using spread spectrum in the spatial domain or compressed domain. Wong *et al.* [9] embeds watermark in the log-2 spatial domain. Lu *et al.* [10] extends cocktail watermark to become a blind multipurpose watermarking system which serves as both robust and fragile watermarks, capable of detecting malicious modifications if the watermark is known. Hernandez *et al.* [11], [12] uses 2-D multipulse amplitude modulation and spread spectrum to embed bit sequences in digital images and develops an optimal detector. Langelaar *et al.* [13] embeds a bit sequence by modifying the energy difference between adjacent blocks. Wong *et al.* [14] uses hash function to embed the watermark in the least significant bit. Zhang *et al.* [15] embeds a watermark pattern by modifying the DC and low-frequency AC coefficients in the DCT domain.

There are also some semi-blind systems that focus on embedding a rotation, scaling, and translation (RST) invariant watermark pattern for watermark detection. Lin *et al.* [16] embeds a watermark in the Fourier-Mellin transform domain. Solachidis *et al.* [17] uses a circularly symmetric watermark in the discrete fourier transform (DFT) domain. Licks *et al.* [18] uses a different kind of circularly symmetric watermark and requires an exhaustive search in the watermark detection. Stankovic *et al.* [19] embeds watermarks by means of a two-dimensional Radon–Wigner distribution with multiple watermark capabilities. All these RST invariant algorithms require the watermark for watermark detection.

While most schemes embed only a single watermark, some allow for multiple watermark embedding [1], [6], [19], [23]. Cox *et al.* [1] assumes the multiple watermarks are close to orthogonal and simply extend the single watermark algorithms to embed them together. Some [19], [23] embed orthogonal watermarks and extend the single watermark algorithms for multiple watermarks.

While many schemes embed watermarks in raw images, only a few embed watermarks in JPEG-compressed images (.jpg files). JPEG is a common file format for digital cameras and the world wide web. Choi *et al.* [20] and Luo *et al.* [21] use inter-block correlation to embed the bit information in selected DCT coefficients in JPEG images by adding or subtracting an offset to the mean value of the neighboring DCT coefficients.

Manuscript received September 7, 2002; revised February 24, 2003. This work was supported by the Research Grants Council of the Hong Kong Special Administrative Region of China (HKUST6028/01E).

The authors are with the Department of Electrical and Electronic Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong, China.

Digital Object Identifier 10.1109/TCSVT.2003.815948

Hartung [8] and Arena *et al.* [22] use spread spectrum to embed watermarks in I-frames, P-frames or B-frames of MPEG-2 compressed video. Essentially, the coding and watermarking of I-frame is the same as that of JPEG. One problem of embedding watermarks in JPEG-compressed images is that the watermarked images need to be JPEG compatible. This implies that all DCT coefficients need to be re-quantized with the same quantization factor after the watermark insertion. The typically small-magnitude watermark can be completely removed in the re-quantization. No existing methods address this problem explicitly.

In this paper, we propose three blind watermarking techniques to embed watermarks in a watermark space. The first proposed technique, called single watermark embedding (SWE), uses two secret keys to embed a meaningful binary logo image in the watermark space, using the spread spectrum technique and some novel features. It does not require the watermark to be orthogonal to the original data, thus allowing bit sequence embedding even in small images. Based on SWE, the second proposed technique, called multiple watermark embedding (MWE), is developed to embed multiple watermarks simultaneously in the same watermark space. Different secret keys are used for different watermarks. We propose solutions for the special case when the secret keys are orthogonal and for the general case when the secret keys are correlated. We show that correlated secret keys can be better than orthogonal keys. The third proposed technique, called iterative watermark embedding (IWE), embeds a single watermark in a JPEG-compressed image to produce another JPEG-compressed image. A novel iterative approach is used to ensure explicitly the existence of the watermark after re-quantization process.

The paper is organized as follows. The proposed SWE and MWE are introduced in Sections II and III, respectively. The proposed IWE is introduced in Section IV. Experimental results and discussions are given in Section V.

II. SWE

In this section, we propose a SWE scheme to embed a single watermark in an image. SWE can be applied in transform domains such as DCT, and possibly the spatial domain, of an image. We group some selected image pixels or transform coefficients to form a vector and call it the watermark host vector. Let the watermark host vector be $\boldsymbol{Y} = [y_1, y_2, \ldots, y_M]$ with length M. The watermark $\boldsymbol{L} = [l_1, l_2, \ldots, l_N]$ with $l_i \in \{0,1\}$ is a bit sequence with length N, where $N \ll M$. The bit sequence may be a meaningful image such as the logo of the image owner or the information related to the host images such as the owner's name, image **ID**, ..., etc. We modulate the watermark by a bit-wise logical XOR operation with a pseudorandom bit sequence $\boldsymbol{S} = [s_1, s_2, \ldots, s_N]$ with $s_i \in \{0,1\}$ to give the modulated watermark sequence $\boldsymbol{W} = [w_1, w_2, \ldots, w_N]$ with $w_i = s_i \oplus l_i$. In the rest of the paper, we will simply use $\boldsymbol{W}$ to represent the modulated watermark.

To embed the watermark, we divide the host vector $\boldsymbol{Y}$ into $\boldsymbol{N}$ subvectors of equal length $P = \lfloor M/N \rfloor$, with the i^{th} subvector denoted as $\boldsymbol{Y_i}$. Each subvector is used to embed one bit of watermark information. SWE uses two keys to embed $\boldsymbol{W}$ into the host vector $\boldsymbol{Y}$ to form the watermarked vector $\boldsymbol{Y'_i}$. The first key, denoted as $\boldsymbol{D} = [d_1, d_2, \ldots, d_N | d_i \in \Re^+ \le i \le N]$, is a set of N pseudorandom positive real numbers. The second key is $\boldsymbol{K} = [k_1, k_2, \ldots, k_M]$ with each k_i being zero-mean Gaussian with variance σ_i. Both keys are needed to decode or detect the modulated watermark. Similar to the host vector $\boldsymbol{Y}$, we split both $\boldsymbol{K}$ and $\boldsymbol{Y'_i}$ into N subvectors of equal length P, with the i^{th} subvector denoted as $\boldsymbol{K_i}$ and $\boldsymbol{Y'_i}$, respectively.

Fig. 1 shows the two-dimensional plane spanned by $\boldsymbol{Y_i}$ and $\boldsymbol{K_i}$. We segment the axis of $\boldsymbol{K_i}$ (or subspace spanned by $\boldsymbol{K_i}$) into disjoint cells of width d_i, and assign the cells alternately to "1" and "0". The i^{th} watermark bit is embedded by adding a small deviation in the direction of $\boldsymbol{K_i}$

$$\boldsymbol{Y}'_i = \boldsymbol{Y}_i + \alpha_i \boldsymbol{K}_i \tag{1}$$

where

$$\alpha_i = \begin{cases} \dfrac{d_i \cdot \text{round}\left(\frac{\langle \boldsymbol{Y}_i, \boldsymbol{K}_i \rangle}{d_i}\right) - \langle \boldsymbol{Y}_i, \boldsymbol{K}_i \rangle}{\|\boldsymbol{K}_i\|_2^2}, & \text{for case 1} \\ \dfrac{d_i \cdot \left(\text{round}\left(\frac{\langle \boldsymbol{Y}_i, \boldsymbol{K}_i \rangle}{d_i}\right) + 1\right) - \langle \boldsymbol{Y}_i, \boldsymbol{K}_i \rangle}{\|\boldsymbol{K}_i\|_2^2}, & \text{for case 2} \\ \dfrac{d_i \cdot \left(\text{round}\left(\frac{\langle \boldsymbol{Y}_i, \boldsymbol{K}_i \rangle}{d_i}\right) - 1\right) - \langle \boldsymbol{Y}_i, \boldsymbol{K}_i \rangle}{\|\boldsymbol{K}_i\|_2^2}, & \text{for case 3} \end{cases} \tag{2}$$

$$\text{Case 1}: \quad \text{round}\left(\frac{\langle \boldsymbol{Y}_i, \boldsymbol{K}_i \rangle}{d_i}\right) \%2 = w_i$$

$$\text{Case 2}: \quad \text{round}\left(\frac{\langle \boldsymbol{Y}_i, \boldsymbol{K}_i \rangle}{d_i}\right) \%2 \neq w_i \quad \text{and } \langle \boldsymbol{Y}_i, \boldsymbol{K}_i \rangle \ge d_i \cdot \text{round}\left(\frac{\langle \boldsymbol{Y}_i, \boldsymbol{K}_i \rangle}{d_i}\right)$$

$$\text{Case 3}: \quad \text{round}\left(\frac{\langle \boldsymbol{Y}_i, \boldsymbol{K}_i \rangle}{d_i}\right) \%2 \neq w_i \quad \text{and } \langle \boldsymbol{Y}_i, \boldsymbol{K}_i \rangle < d_i \cdot \text{round}\left(\frac{\langle \boldsymbol{Y}_i, \boldsymbol{K}_i \rangle}{d_i}\right)$$

where round( ), %, and $\|\cdot\|_2$ are rounding, modulo 2 and L^2-norm respectively, and $\langle \boldsymbol{Y}_i, \boldsymbol{K}_i \rangle$ is the inner product of $\boldsymbol{Y}_i$ and $\boldsymbol{K}_i$. In case 1, the projection of $\boldsymbol{Y}_i$ onto the $\boldsymbol{K}_i$-axis falls in a cell of the desired watermark bit. In cases 2 and 3, the projection falls in a wrong cell. In case 2, the cell on the right is closer. In case 3, the cell on the left is closer. We force the projection of $\boldsymbol{Y'_i}$ to be at the center of the nearest cell of the desired watermark bit. The nearest cell would ensure minimal distortion to the image. The center of the cell would give maximum robustness because the distance to the nearest cell boundary is maximum at the center of a cell. It would take a distortion of at least $d_i/2$ in the direction of $\boldsymbol{K_i}$ to incur an error in the i^{th} decoded bit. In this sense, the cell width d_i controls the robustness of the embedded watermark bit. However, a large d_i may lead to visible artifacts as the distortion due to watermarking is larger than $d_i/2$ in cases 2 and 3.

To decode the watermark bits for SWE, we extract the watermark vector $\boldsymbol{Y'}$ from the test image and segment it into N subvectors of length P. The i^{th} modulated watermark bit w'_i is decoded from the i^{th} subvector $\boldsymbol{Y}'_i$ as $w'_i = \text{round}(\langle \boldsymbol{Y}'_i, \boldsymbol{K}_i \rangle / d_i) \%2$ using the two keys and the i^{th} demodulated watermark bit is obtained as $l'_i = w'_i \oplus s_i$.

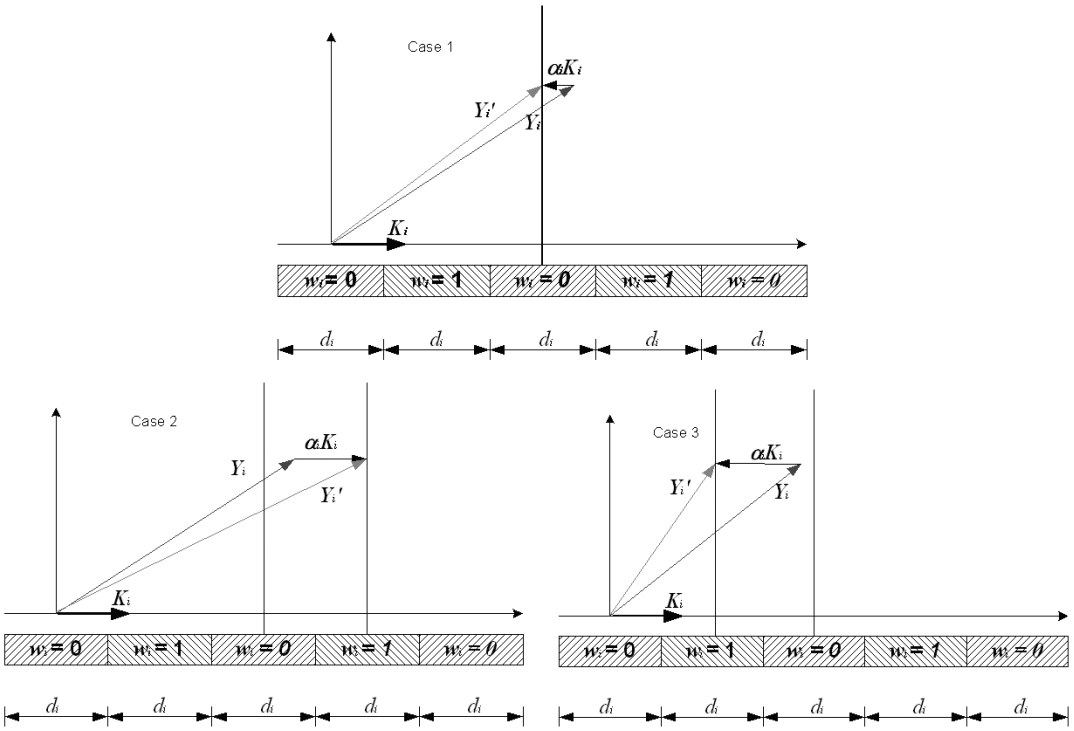

Fig. 1. Modification of subvector in SWE watermark embedding.

To detect whether a modulated watermark $\boldsymbol{W} = [w_1, w_2, \ldots, w_N]$ is present in a testing image, we decode all the N bits of watermark from the image as $\boldsymbol{W'} = [w'_1, w'_2, \ldots, w'_N]$ and evaluate a score. A possible score is the traditional normalized cross correlation S_1 between the original watermark and the decoded watermark

$$S_1 = \frac{\sum_{i=1}^{N}(2 \cdot w_i - 1) \cdot (2 \cdot w'_i - 1)}{\sqrt{\sum_{i=1}^{N}(2 \cdot w_i - 1)^2 \cdot \sum_{i=1}^{N}(2 \cdot w'_i - 1)^2}} = \frac{1}{N}\sum_{i=1}^{N}(2 \cdot w_i - 1) \cdot (2 \cdot w_i{}' - 1). \quad (3)$$

Each watermark bit is transformed from $\{0, 1\}$ to $\{-1, 1\}$ in (3). When SWE is applied in the transform domain, another possible score is the weighted normalized cross correlation score S_2

$$S_2 = \frac{\sum_{i=1}^{N}[\beta_i \cdot (2 \cdot w_i - 1)] \cdot [\beta_i \cdot (2 \cdot w'_i - 1)]}{\sqrt{\sum_{i=1}^{N}[\beta_i \cdot (2 \cdot w_i - 1)]^2 \cdot \sum_{i=1}^{N}[\beta_i \cdot (2 \cdot w'_i - 1)]^2}} = \sum_{i=1}^{N}\frac{\beta_i^2}{\sum_{j=1}^{N}\beta_j^2} \cdot (2 \cdot w_i - 1) \cdot (2 \cdot w'_i - 1) \quad (4)$$

where $\beta_i \geq 0$. We choose relatively large β_i for the watermark bits embedded in the low-frequency components of the image and smaller β_i for the high frequency components, because the decoded bits in the low-frequency components tend to be more trustworthy. Attacks (intentional or unintentional) on low-frequency components tend to be less severe than in high frequency because distortions in the low-frequency components tend to be more visible. If the detection score is higher than a pre-defined threshold, the watermark is considered to be present in the testing image. Note that the use of the weighting factor β_i does not affect the embedding process.

III. MWE

In this section, we generalize SWE to embed multiple watermarks in the same image while retaining high image quality. In SWE, we embed only one bit in each subvector. In MWE, we embed Q bits simultaneously in each subvector $\boldsymbol{Y_i}$. We generalize the first key to Q sets of N pseudorandom positive real numbers denoted as $\boldsymbol{D_1}, \boldsymbol{D_2}, \ldots, \boldsymbol{D_Q}$ and the second key to Q pseudorandom vectors of length M, denoted as $\boldsymbol{K_1}, \boldsymbol{K_2}, \ldots, \boldsymbol{K_Q}$ with $\boldsymbol{K}_j = \lfloor k_{j,1}, k_{j,2}, \ldots, k_{j,M} \rfloor$, for $1 \leq j \leq Q$. Similar to SWE, we split the host vector $\boldsymbol{Y}$ and the random vectors $\boldsymbol{K_1}, \boldsymbol{K_2}, \ldots, \boldsymbol{K_Q}$ into N subvectors of length P. The i^{th} element of $\boldsymbol{D}_j$ is denoted as $d_{j,i}$ and the i^{th} subvector of $\boldsymbol{K}_j$ is denoted as $\boldsymbol{K}_{j,i}$. The i^{th} bit of the j^{th} watermark sequence is denoted as $w_{j,i}$. The watermarked i^{th} subvector, denoted as $\boldsymbol{Y}'_i$, is

$$\boldsymbol{Y}'_i = \boldsymbol{Y}_i + \alpha_{1,i}\boldsymbol{K}_{1,i} + \alpha_{2,i}\boldsymbol{K}_{2,i} + \ldots + \alpha_{Q,i}\boldsymbol{K}_{Q,i}. \quad (5)$$

The scaling factors form a row vector $\boldsymbol{A}_i = \lfloor \alpha_{1,i}, \alpha_{2,i}, \ldots, \alpha_{Q,i} \rfloor$ with $\alpha_{j,i} \in \Re$.

The goal of the watermark embedding process is to derive a set of scaling factors (or vector $\boldsymbol{A}_i$) which satisfies two conditions. The first condition is that the projection of $\boldsymbol{Y}'_i$ onto the direction of $\boldsymbol{K}_{j,i}$ corresponds to the correct watermark bit as

$$\left[\text{Round}\left(\frac{\langle \boldsymbol{Y}'_i, \boldsymbol{K}_{j,i} \rangle}{d_{j,i}}\right)\right] \%2 = w_{j,i}, \quad 1 \leq j \leq Q. \quad (6)$$

The second condition is that the distortion or the Euclidean distance E_w between $\boldsymbol{Y}_i$ and $\boldsymbol{Y}_i'$ is minimized. The Euclidean distance E_w is also the energy of the watermark and is equal to

$$E_w = \|\boldsymbol{Y}_i{}' - \boldsymbol{Y}_i\|_2^2 = \boldsymbol{A}_i \boldsymbol{C}_i \boldsymbol{A}_i^T \tag{7}$$

where

$$\boldsymbol{C}_i = \begin{pmatrix} \|\boldsymbol{K}_{1,i}\| & \langle \boldsymbol{K}_{1,i}, \boldsymbol{K}_{2,i}\rangle & \cdots\cdots & \langle \boldsymbol{K}_{1,i}, \boldsymbol{K}_{Q,i}\rangle \\ \langle \boldsymbol{K}_{2,i}, \boldsymbol{K}_{1,i}\rangle & \|\boldsymbol{K}_{2,i}\| & \cdots\cdots & \langle \boldsymbol{K}_{2,i}, \boldsymbol{K}_{Q,i}\rangle \\ \vdots & \ddots & \ddots\ \ddots & \vdots \\ \vdots & \ddots & \ddots\ \ddots & \vdots \\ \langle \boldsymbol{K}_{Q,i}, \boldsymbol{K}_{1,i}\rangle & \langle \boldsymbol{K}_{Q,i}, \boldsymbol{K}_{2,i}\rangle & \cdots\cdots & \|\boldsymbol{K}_{Q,i}\| \end{pmatrix} \tag{8}$$

Substituting (5) into (6), Q simultaneous equations can be obtained. They are, in matrix form

$$\boldsymbol{A}_i \boldsymbol{C}_i = \boldsymbol{B}_i = \lfloor b_{1,i}, b_{2,i}, \ldots, b_{Q,i} \rfloor \tag{9}$$

with

$$b_{j,i} = d_{j,i} \cdot (2 \cdot r_{j,i} + w_{j,i}) + \langle \boldsymbol{Y}_i, \boldsymbol{K}_{j,i}\rangle, \quad 1 \le j \le Q \tag{10}$$

where $r_{j,i}$ is an integer for any i, j. If all the $r_{j,i}$ are determined, the row vector $\boldsymbol{B}_i$ can be computed using (10) and the scaling vector $\boldsymbol{A}_i$ can then be obtained as $\boldsymbol{A}_i = \boldsymbol{B}_i \boldsymbol{C}_i^{-1}$ from (9). It is important to choose the integers $r_{j,i}$ such that the E_w is as small as possible. By substituting (9) into (7), the E_w can be rewritten in terms of $\boldsymbol{C}_i$ and $\boldsymbol{B}_i$ as

$$E_w = \boldsymbol{B}_i \boldsymbol{C}_i^{-1} \boldsymbol{B}_i^T \tag{11}$$

We will now propose two approaches to choose the $r_{j,i}$. The first approach is called the *direct approach (DA)*, which is simple and is optimal for the special case of orthogonal random key vectors. The second approach is called *iterative approach (IA)*, which is more suitable for the general case of nonorthogonal random key vectors. The second approach is useful because nonorthogonal random vectors can potentially incur smaller host signal distortion than orthogonal ones. This can be seen in the example in Figs. 2 and 3, which have the same $d_{1,i}$ and $d_{2,i}$ when $Q = 2$. The random vectors are orthogonal in Fig. 3 but are not in Fig. 2. The distortion incurred between $\boldsymbol{Y}_i$ and $\boldsymbol{Y}_i'$ is smaller in Fig. 2 than in Fig. 3.

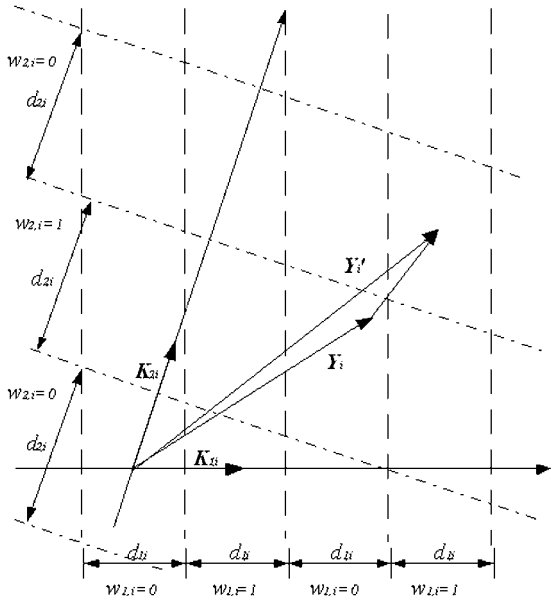

Fig. 2. Modification of subvector using correlated random subvectors.

A. The DA

In our DA, $r_{j,i}$ is chosen according to (12) to minimize the absolute value of $b_{j,i}$, as shown in (12), at the bottom of the page. It is easy to show that (12) guarantees all $r_{j,i}$ are integers. The advantage of using DA is its simplicity. And this is the optimal solution for the special case of orthogonal random key vectors. It is not optimal in the general case of nonorthogonal key vectors.

$$r_{ij} = \begin{cases} \frac{1}{2} \cdot \left[\text{Round}\left(\frac{\langle \boldsymbol{Y}_i, \boldsymbol{K}_{j,i}\rangle}{d_{j,i}} \right) - w_{j,i} \right], & \text{for case 1} \\ \frac{1}{2} \cdot \left[\text{Round}\left(\frac{\langle \boldsymbol{Y}_i, \boldsymbol{K}_{j,i}\rangle}{d_{j,i}} \right) - w_{j,i} + 1 \right], & \text{for case 2} \\ \frac{1}{2} \cdot \left[\text{Round}\left(\frac{\langle \boldsymbol{Y}_i, \boldsymbol{K}_{j,i}\rangle}{d_{j,i}} \right) - w_{j,i} - 1 \right], & \text{for case 3} \end{cases} \tag{12}$$

$$\text{Case 1}: \quad \text{round}\left(\frac{\langle \boldsymbol{Y}_i, \boldsymbol{K}_{j,i}\rangle}{d_{j,i}} \right) \%2 = w_{j,i}$$

$$\text{Case 2}: \quad \text{round}\left(\frac{\langle \boldsymbol{Y}_i, \boldsymbol{K}_{j,i}\rangle}{d_{j,i}} \right) \%2 \neq w_{j,i} \quad \text{and } \langle \boldsymbol{Y}_i, \boldsymbol{K}_{j,i}\rangle \ge d_{j,i} \cdot \text{round}\left(\frac{\langle \boldsymbol{Y}_i, \boldsymbol{K}_{j,i}\rangle}{d_{j,i}} \right)$$

$$\text{Case 3}: \quad \text{round}\left(\frac{\langle \boldsymbol{Y}_i, \boldsymbol{K}_{j,i}\rangle}{d_{j,i}} \right) \%2 \neq w_{j,i} \quad \text{and } \langle \boldsymbol{Y}_i, \boldsymbol{K}_{j,i}\rangle < d_{j,i} \cdot \text{round}\left(\frac{\langle \boldsymbol{Y}_i, \boldsymbol{K}_{j,i}\rangle}{d_{j,i}} \right).$$

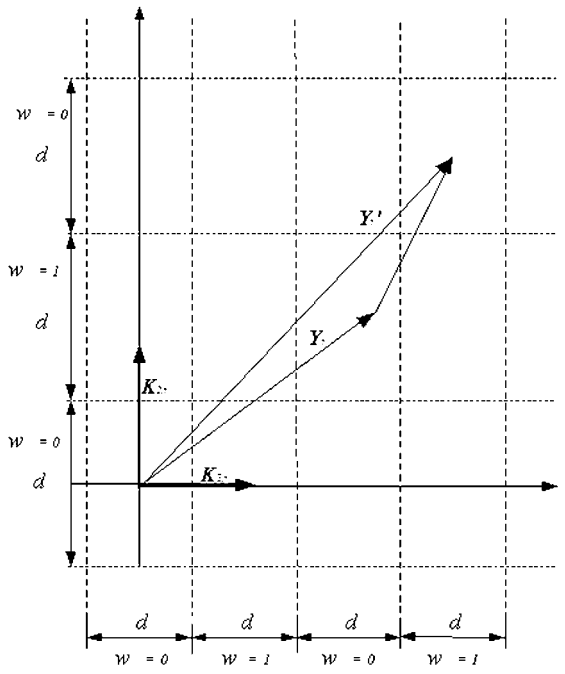

Fig. 3. Modification of subvector using orthogonal random subvectors.

B. The IA

Here is the iterative approach. Expanding (11), Ew can be written as a second-order function of $b_{1,i}, b_{2,i}, b_{3,i}, \ldots, b_{Q,i}$

$$E_w = \sum_{k=1}^{Q} C_i^{-1}(k,k) \cdot b_{k,i}^2 + 2 \cdot \sum_{k=1}^{Q-1} \sum_{l=k+1}^{Q} C_i^{-1}(k,l) \cdot b_{k,i} \cdot b_{l,i} \tag{13}$$

where $C_i^{-1}(k,l)$ is the $(k,l)^{\text{th}}$ element of matrix $\boldsymbol{C}_i^{-1}$. The trivial solution $b_{1,i} = b_{2,i} = \ldots = b_{Q,i} = 0$ achieves minimum E_w. However, these values are invalid as they are not in the form of (10). We need to find a valid $\boldsymbol{B}_i$ in the form of (10), where all $r_{j,i}$ are integers, such that E_w is minimized.

In the proposed IA, we maintain a pool of valid candidate vectors for $\boldsymbol{B}_i$ and will allow the size of the pool to grow as the iterations proceed. The algorithm starts with a vector pool with only one valid vector $\boldsymbol{B}_i$ obtained from the DA. In each iteration, each valid vector in the pool will be optimized by an algorithm to be described later to yield some new valid vectors. If any new valid vector is not already in the valid candidate vector pool, it will be added to it for the next iteration. The E_w is computed for all the vectors in the pool and the minimum E_w is identified. The iterations will stop when the incremental reduction of the minimum E_w is less than a threshold.

Here is the two-stage optimization done in each iteration. In the first stage, consider any valid vector $\boldsymbol{B_i}$ in the vector pool. A set of vectors is produced from $\boldsymbol{B_i}$ by optimizing some elements in $\boldsymbol{B_i}$ while keeping the others unchanged. Let the number of elements to be optimized be N_A. Since there are ${}_QC_{N_A} = Q!/[(Q-N_A)! \cdot N_A!]$ ways to choose N_A elements from $\boldsymbol{B_i}$ (of length Q), there are ${}_QC_{N_A}$ optimized vectors produced from each vector $\boldsymbol{B_i}$ after the first stage. We find experimentally that $\lfloor Q/2 \rfloor$ or $\lfloor Q/2 \rfloor + 1$ are good values for N_A. Let $\boldsymbol{S}_A$ be the set of the index of the N_A elements selected from $\boldsymbol{B_i}$. Then E_w can be written as

$$\begin{aligned} E_w = & \sum_{j \in \boldsymbol{S_A}} C_i^{-1}(j,j) \cdot b_{j,i}^2 + \sum_{j,k \in \boldsymbol{S_A}, j \neq k} C_i^{-1}(j,k) \cdot b_{j,i} \cdot b_{k,i} \\ & + \sum_{j \in \boldsymbol{S_A}, k \notin \boldsymbol{S_A}} \left[C_i^{-1}(j,k) + C_i^{-1}(k,j) \right] \cdot b_{j,i} \cdot b_{k,i} \\ & + \sum_{j \in \boldsymbol{S_A}} C_i^{-1}(j,j) \cdot b_{j,i}^2 \\ & + \sum_{j,k \in \boldsymbol{S_A}, j \neq k} C_i^{-1}(j,k) \cdot b_{j,i} \cdot b_{k,i}. \end{aligned} \tag{14}$$

Differentiating E_w with respect to each element of S_A and setting the derivatives to be zero, we get N_A simultaneous equations

$$\sum_{j \in \boldsymbol{S_A}} C_i^{-1}(j,k) \cdot b_{j,i} = - \sum_{j \notin \boldsymbol{S_A}} C_i^{-1}(j,k) \cdot b_{j,i}, \quad \text{for } k \in \boldsymbol{S_A} \tag{15}$$

Since E_w is a quadratic function and positive definite, E_w attains its minimum when (15) is satisfied. By solving (15), one optimal vector of $\boldsymbol{B}_i$ is obtained.

Typically, the optimal vectors produced in the first stage are not valid since they are not in the form of (10) with integer $r_{j,i}$. The second stage is now applied to modify these optimal vectors to produce valid suboptimal vectors. Consider a particular optimal vector $\boldsymbol{B}_{i_{\text{opt}}}$. Our problem is that the corresponding $r_{j,i_{\text{opt}}}$ computed from (10)

$$r_{j,i_{\text{opt}}} = \frac{\frac{\left(b_{j,i_{\text{opt}}} + \langle \boldsymbol{Y}_i, \boldsymbol{K}_{j,i} \rangle\right)}{d_{j,i}} - w_{j,i}}{2}, \quad \text{for } j \in \boldsymbol{S_A} \tag{16}$$

is not an integer. We use two methods to convert it into an integer. The first is to simply round $r_{j,i_{\text{opt}}}$ to the nearest integer. We call this IA-rounding or IA-R in short. The second way is to include all the combinations of floor $\lfloor r_{j,i_{\text{opt}}} \rfloor$ and ceiling, $\lceil r_{j,i_{\text{opt}}} \rceil$ for $j \in \boldsymbol{S_A}$. This will produce N_A^2 suboptimal valid vectors. We call this IA-full or IA-F in short. The pool size should grow faster in IA-F than IA-R.

C. Watermark Decoding and Detection

Although multiple watermarks are embedded in the host signal simultaneously in MWE, each embedded watermark bit sequence can be decoded independently, as in SWE. Watermark detection is performed in a similar way by decoding the watermark bit sequence and computing the scores S_1 or S_2 in Section II-A.

IV. IWE in JPEG-Compressed Domain

An interesting problem arises when the original image for the proposed watermarking algorithm is a JPEG-compressed image from a .jpg file and the watermarked image needs to be JPEG recompressed to produce another .jpg file. Would the watermark still be decodable or detectable in the JPEG recompressed file? Given that the original image is JPEG-compressed, the original DCT coefficients are already quantized. For any watermarking methods, the distortion of these quantized DCT coefficients due

to watermarking is typically small compared with the quantization factor. During the recompression to produce the JPEG-compatible file, the watermarked DCT coefficients are re-quantized and this can completely remove the small distortion which carries the watermark information and restore the original quantized DCT values. This is especially serious when the compression ratio is large. Based on SWE, we propose a novel IWE method to prevent the removal of the watermark in the requantization process such that watermark decoding and detection can still work.

IWE assumes that the watermarked image will be recompressed with the same quantization matrix and quantization factor as the original JPEG-compressed image. Initially, the watermark host vector $\boldsymbol{Y}$ of length M is extracted directly from the JPEG compressed domain of the original image. As the original image is already JPEG-compressed, the image is partitioned into 8×8 blocks and the DCT coefficients are arranged in zigzag order in the original .jpg file. To form the watermark host vector, the first AC coefficient in zigzag order is extracted from all the 8×8 blocks to form the first portion of $\boldsymbol{Y}$. Then the second AC coefficient in zigzag order is extracted from the blocks and appended to form the second portion of $\boldsymbol{Y}$, and so on until a total of M coefficients are extracted. We divide the host vector $\boldsymbol{Y}$ into N subvectors of P.

Here is the proposed IWE for any subvector $\boldsymbol{Y_i}$. For any $\boldsymbol{Y_i}$, a random noise subvector $\boldsymbol{N_i}$ of length P is generated and added to $\boldsymbol{Y_i}$ before SWE is applied. Each element of the random noise is uniformly distributed in $[-q/2, q/2]$, where q is the effective quantization cell width of the corresponding DCT coefficient. This is similar to dithering. If the original quantized DCT coefficient is zero, the random noise element is forced to be zero. After the watermark embedding, JPEG requantization (with same quantization matrix and scaling factor as in the original image) is performed followed by watermark decoding. If the watermark bit (or bits) embedded can be decoded correctly, the goal is achieved and IWE stops. Otherwise, another iteration is carried out with another random noise subvector $\boldsymbol{N_i}$ generated and the other steps repeated. To limit the complexity, the iteration is forced to terminate when the number of iterations exceeds a pre-defined threshold. Note that the random noise subvector $\boldsymbol{N_i}$ is not needed in watermark decoding.

After IWE, the altered (watermarked) vector $\boldsymbol{Y_i'}$ is inserted back to the original location. JPEG-requantization (with the same quantization factor and matrix as original) is carried out followed by variable-length coding to generate the final .jpg file.

Fig. 4. Testing images used in the experiments.

Fig. 5. Original 512×512 "Lena'".

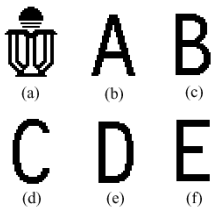

Fig. 6. (a) Original logo "UST". (b)-(f) Original logo "Alphabet".

V. Experimental Results and Discussion

We tested the proposed algorithms on many testing images as shown in Fig. 4. All the images are 512×512 pixels (e.g., see Fig. 5) and only the luminance components are used. Six 44×30 binary logo images, as shown in Fig. 6, are used as perceptual meaningful watermarks in the experiments. The binary images are raster-scanned to form 1-dimensional bit sequences and modulated with a pseudorandom binary sequence. We simulate SWE and MWE in the DCT domain. The whole original image is transformed to the (512×512) DCT domain and scanned in a zigzag order. We use the first 10% of the DCT AC coefficients to form the host vector to embed the watermark. The length of the host vector $\boldsymbol{Y}$ is $M = floor(512 \times 512 \times 0.1) = 26\,214$. The length of the watermark $\boldsymbol{W}$ is $N = 44 \times 30 = 1320$. The length of the subvector $\boldsymbol{Y_i}$ is $P = floor(26214/1320) = 19$. We choose low-frequency components of DCT to form the host vector as these components tend to have large energies such that the embedded watermarks tend to be robust against different kinds of attack.

For the score S_2 in (4), we choose the weighting factor β_i as follows. The default 8×8 quantization matrix in JPEG is bilinear interpolated to 512×512. Treating this as an image,

Fig. 7. SWE-watermarked image, $\mathrm{PSNR} = 46.12$ dB.

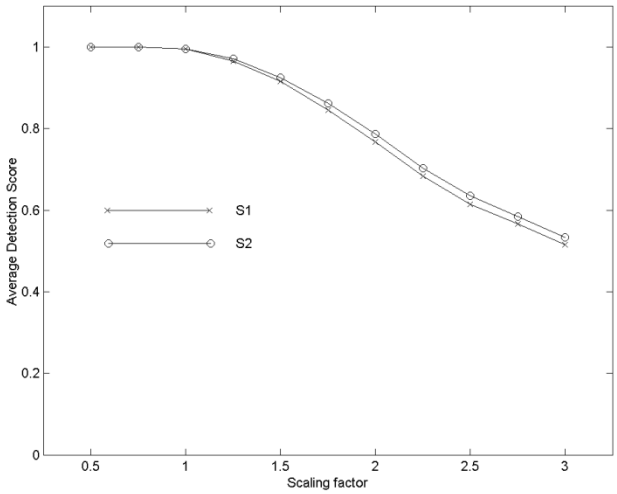

Fig. 8. Average detection score against JPEG compression for SWE.

we extract the host vector of length M and divide them into N subvectors of length P. The weighting factor β_i is chosen as the inverse of the sum of elements of the i^{th} subvector such that the low-frequency components have relatively large β_i.

A. Results of SWE

The proposed SWE is used to embed the "UST" logo in all the testing images. A typical SWE-watermarked image is shown in Fig. 7. With very high PSNR, the watermarked images have very good visual quality. As expected, all the watermark bits can be decoded perfectly under no attack resulting in both detection scores S_1 and S_2 being 1. Several unintentional attacks are simulated, including JPEG compression, low-pass filtering (LPF), noise, and print-and-scan. In the JPEG compression attack, the watermarked images are JPEG compressed with the default quantization matrix scaled by various scaling factor (SF) to achieve different compression ratio. Most research papers control the JPEG quality using the quality factor (QF) and the SF is related to the QF by

$$\mathrm{SF} = \begin{cases} \frac{50}{\mathrm{QF}}, & \mathrm{QF} \leq 50 \\ 2 - \frac{\mathrm{QF}}{50}, & 50 < \mathrm{QF} < 100. \end{cases} \tag{17}$$

JPEG decompression is performed followed by watermark detection. Ten trials with different keys $\boldsymbol{D}$ and $\boldsymbol{K}$ are performed for each SF to obtain the average detection scores. The first key $\boldsymbol{D}$ is a vector of 1320 random numbers generated independently from a Gaussian distribution with $\mathrm{mean} = 480$ and $\mathrm{variance} = 4$. The other key $\boldsymbol{K}$ is generated from a Gaussian distribution with $\mathrm{mean} = 0$ and $\mathrm{variance} = 16$. These parameters are chosen to achieve a PSNR of about 45 dB for the watermarked images. We observe that at such PSNR, the watermarked images are almost indistinguishable from the original image to the human eyes.

The typical average detection scores, S_1 and S_2, are shown in Fig. 8. We observe that, in most cases, S_2 is larger than S_1. This agrees with our finding in the Appendix that the expected value of S_2 is larger than that of S_1, in the case that β_i takes on only two values, a larger one for the bits with low probability of error and vice versa. Perhaps S_2 can be as good as, if not better than, S_1 for watermark detection.

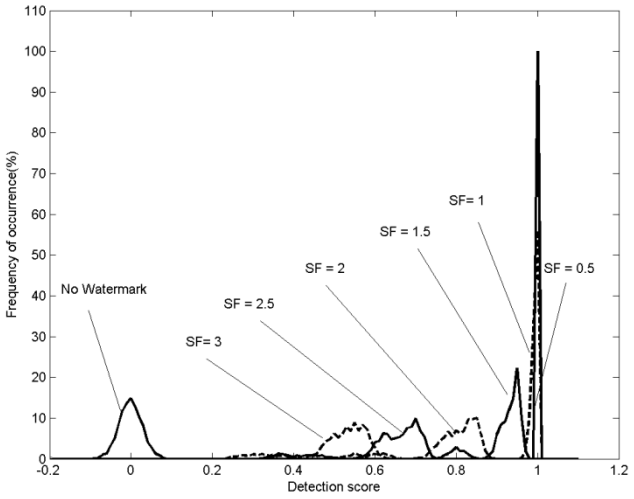

Fig. 9. Distribution of S_1 of SWE under JPEG attack.

The sample distribution of S_1 in SWE watermark detection are shown in Figs. 9, 11, and 13 for JPEG attack, LPF, and noise attack, respectively. There are two types of detection errors for a given detection threshold: type 1 being the false positive error and type 2 being the false negative error. The total detection errors (sum of type 1 and type 2 error probability) against different thresholds are shown in Figs. 10, 12, and 14 respectively. All curves are averages of 100 trials with different keys.

Six situations of JPEG compression attack with $\mathrm{SF} = 0.5$, 1, 1.5, 2, 2.5, and 3 on the SWE-watermarked images are shown in Fig. 9, together with the reference "no watermark" situation. The sample distribution of "no watermark" does not intersect with any of the sample distributions of the six JPEG situations. As a result, any threshold values in the "in-between" region in Fig. 10 can give zero total detection error. A typical example of SWE under JPEG attack is shown in Fig. 15. At $\mathrm{SF} = 3.0$ (0.319 bpp) with a PSNR of 32.23 dB, the JPEG-compressed image has rather severe and visible image distortion due to JPEG compression. The severe compression attack causes the decoded watermark (the 44×30 binary logo) to be noisy with a bit error

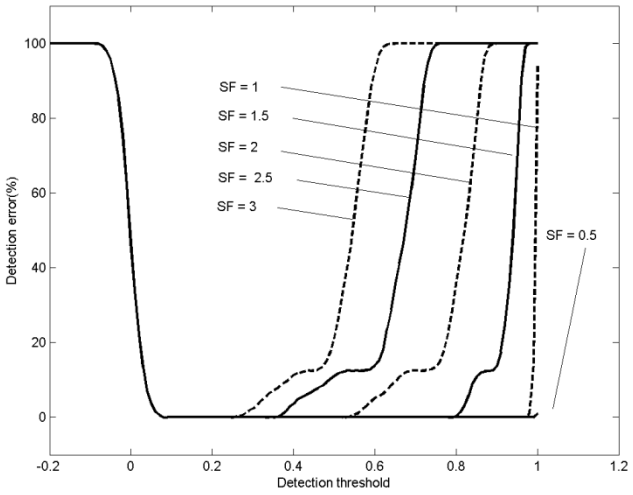

Fig. 10. Detection error of SWE under JPEG attack.

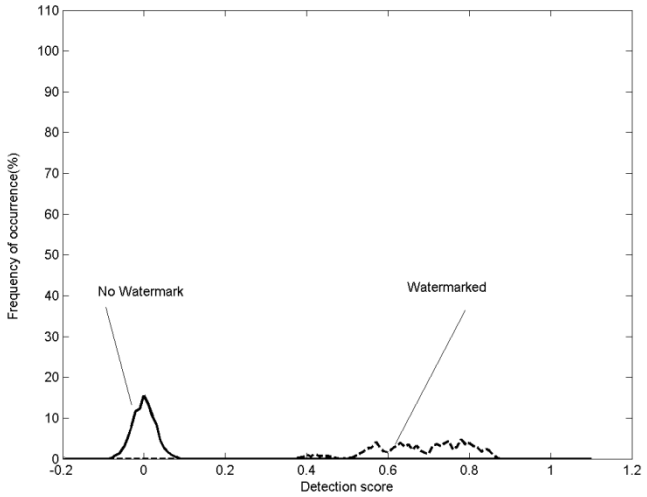

Fig. 11. Distribution of S_1 of SWE under LPF attack.

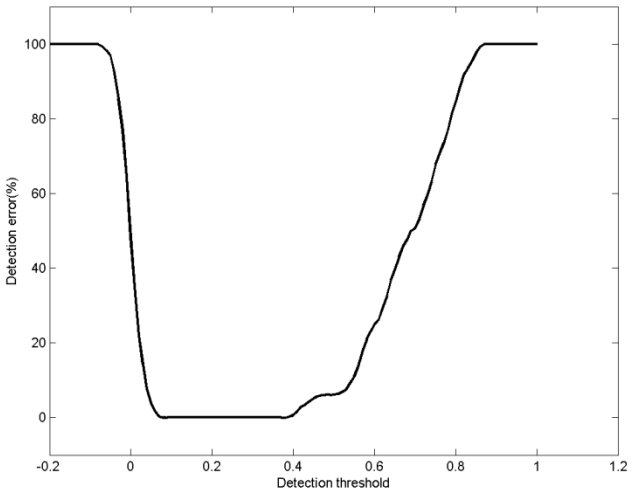

Fig. 12. Detection error of SWE under LPF attack.

rate of 25.08%. Although the decoded 'UST' logo is barely visible, the detection scores remain quite large at $S_1 = 0.4985$ and $S_2 = 0.5208$. The watermark can be clearly distinguished in the random watermark test in Fig. 15.

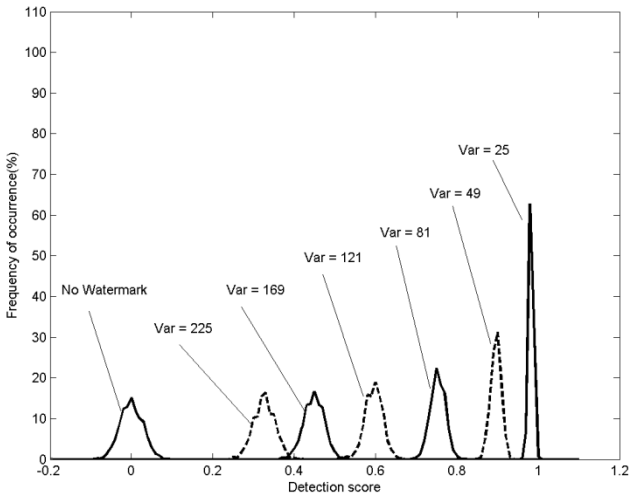

Fig. 13. Distribution of S_1 of SWE under noise attack.

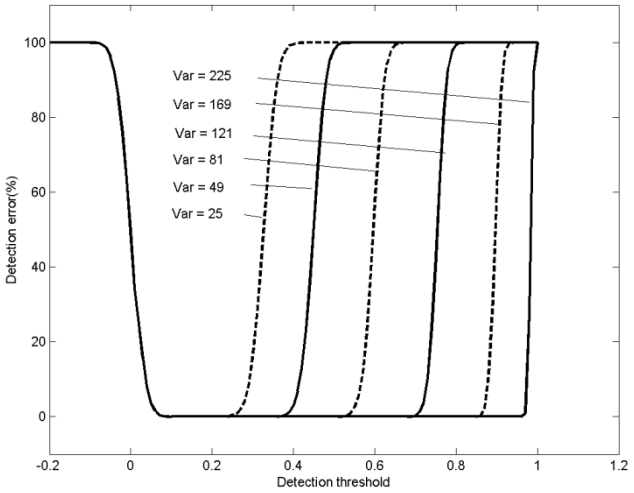

Fig. 14. Detection error of SWE under noise attack.

Similarly, the sample distributions of "watermarked" and "no watermark" under LPF attack do not intersect in Fig. 11, and thus there are many thresholds in Fig. 12 that can give zero detection error. A 3×3 averaging filter (with coefficients of 1/9) is used in the LPF. A typical example of SWE under LPF attack is shown in Fig. 16. The LPF attacked image is blurred with a PSNR of 31.85 dB. The LPF attack causes the decoded watermark to be somewhat noisy with a bit error rate of 14.62%. Although the "'UST" logo is visibly noisy, the detection scores remain large at $S_1 = 0.7076$ and $S_2 = 0.7150$. The watermark is clearly distinguishable in the random watermark test.

Six situations of Gaussian noise attack with different noise variance (Var) are shown in Fig. 13, together with the 'no watermark' situation. Again, the nonintersection of 'no watermark' and other sample distributions lead to the possibility of zero detection error in Fig. 14. A typical example of SWE under Gaussian noise attack is shown in Fig. 17. The image attacked by an additive zero-mean Gaussian noise with variance 225 is noisy with a PSNR of only 24.26 dB. The severe noise causes the decoded watermark to be very noisy with a bit error rate of 32.35%. Although the "UST" logo can hardly be recognized and the detection scores are low at $S_1 = 0.3530$ and $S_2 = 0.3537$,

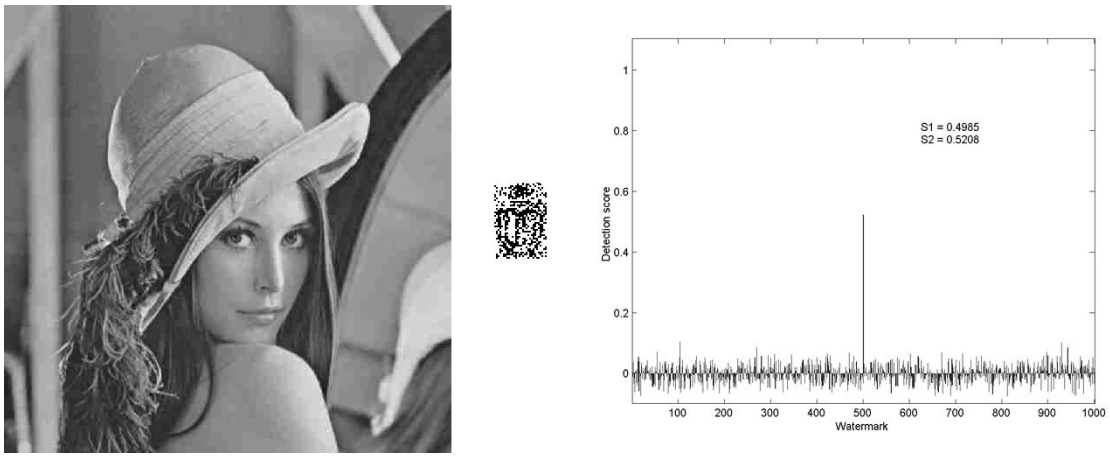

Fig. 15. (Left) SWE-watermarked image under JPEG attack, $\mathrm{SF} = 3$, $\mathrm{PSNR} = 32.23$ dB, $\mathrm{bpp} = 0.319$. (Middle) Decoded watermark. (Right) Random watermark detection results.

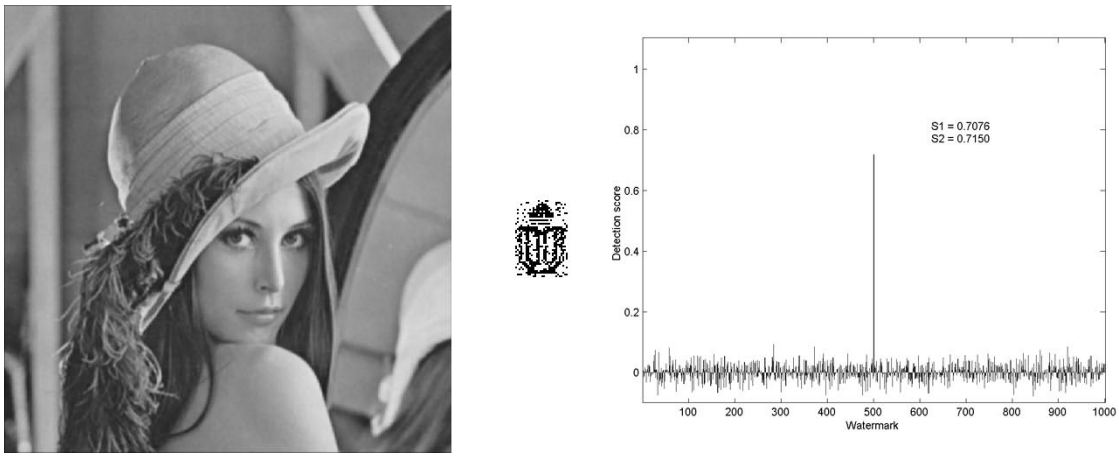

Fig. 16. (Left) SWE-watermarked image under LPF attack, $\mathrm{PSNR} = 31.85$ dB. (Middle) Decoded watermark. (Right) Random watermark detection results.

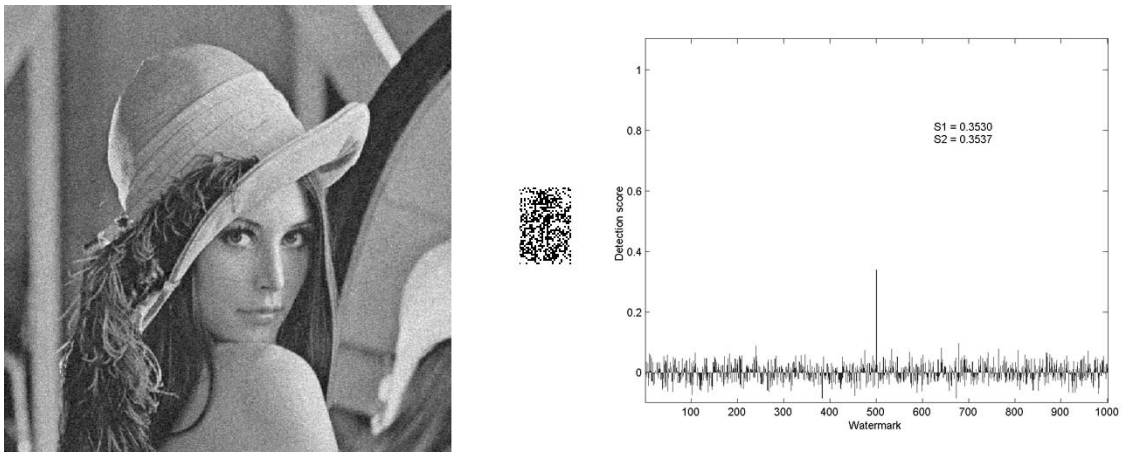

Fig. 17. (Left) SWE-watermarked image under noise attack, $\mathrm{PSNR} = 24.26$ dB, $\mathrm{variance} = 225$. (Middle) Decoded watermark. (Right) Random watermark detection results.

the watermark is still distinguishable in the random watermark test.

In the print-and-scan attack, the watermarked images are printed at 180 dpi (2.84 inch $\times$ 2.84 inch) on photo papers using a 2880dpi Epson 895 printer. They are then scanned with a 2400 dpi Epson 1250 scanner resulting in images with approximately 7000 $\times$ 7000 pixels. Software (e.g., Photoshop) is used to rotate manually the scanned images to an upright position, to crop out the image regions and to resample them down to 512 $\times$ 512 images using bi-cubic interpolation. A typical example of the print-and-scan attack is shown in Fig. 18. The resampled image at a PSNR of 24.79 dB looks quite good.

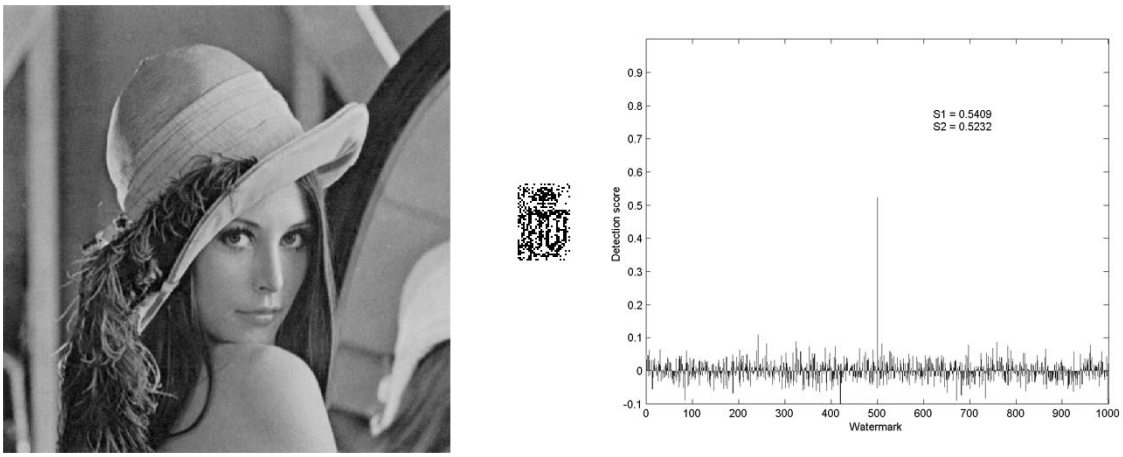

Fig. 18. (Left) SWE-watermarked image under print-and-scan attack, $\mathrm{PSNR} = 24.79$. (Middle) Decoded watermark. (Right) Random watermark detection results.

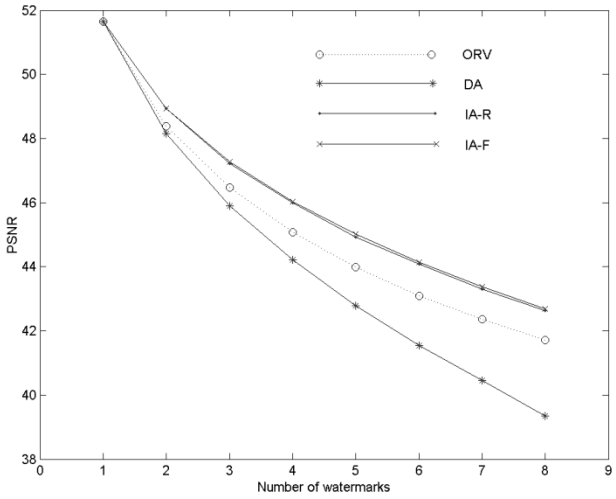

Fig. 19. Average PSNR of MWE versus number of watermarks (Q).

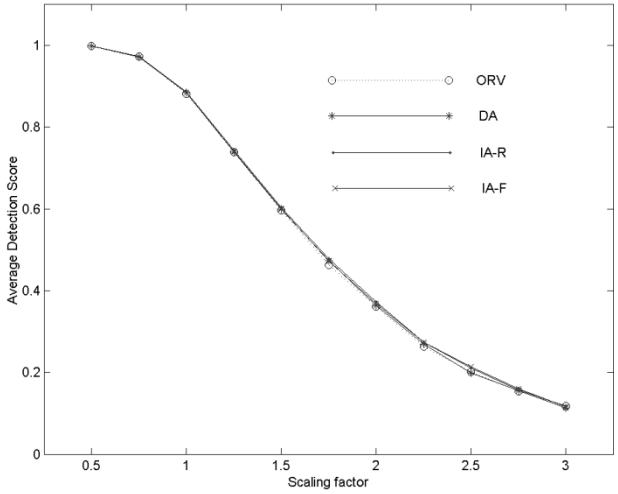

Fig. 21. Average detection score S_1 of MWE $(\mathrm{Q} = 5)$ versus JPEG compression SF.

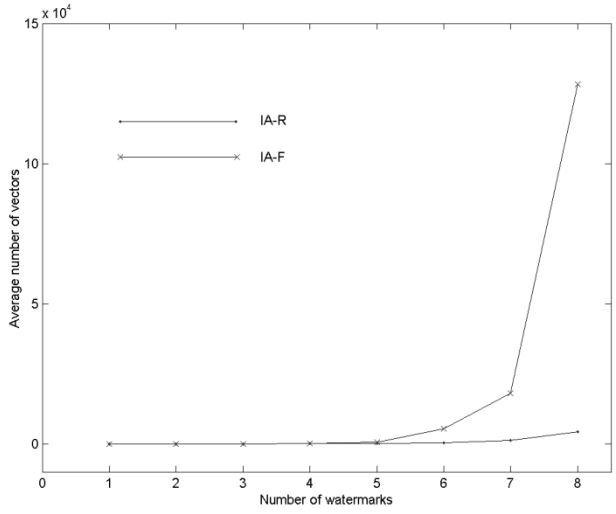

Fig. 20. Complexity comparison between IA-R and IA-F.

Fig. 22. MWE-watermarked image with five watermarks embedded ($\mathrm{Q} = 5$, IA-R). $\mathrm{PSNR} = 44.94$ dB.

The low PSNR suggests that the manual rotation and cropping might have led to misalignment. The print-and-scan attack causes the decoded watermark to be very noisy with a bit error rate of 22.95%. Although the "UST" logo is noisy, the detection scores are quite large at $S_1 = 0.5409$ and $S_2 = 0.5232$. The watermark is distinguishable in the random watermark test results. In this case, S_1 is larger than S_2.

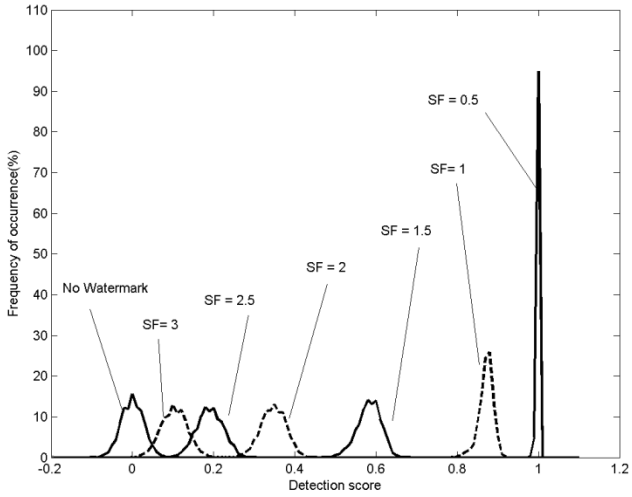

Fig. 23. Distribution of S_1 of MWE (IA-R, $\mathrm{Q} = 5$) under JPEG attack.

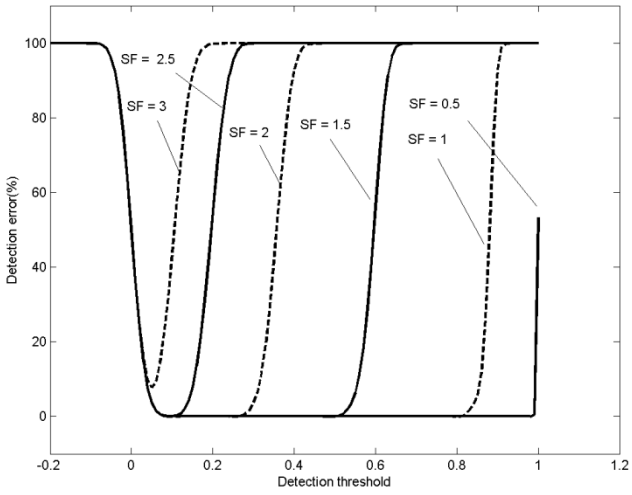

Fig. 24. Detection error of MWE under JPEG attack.

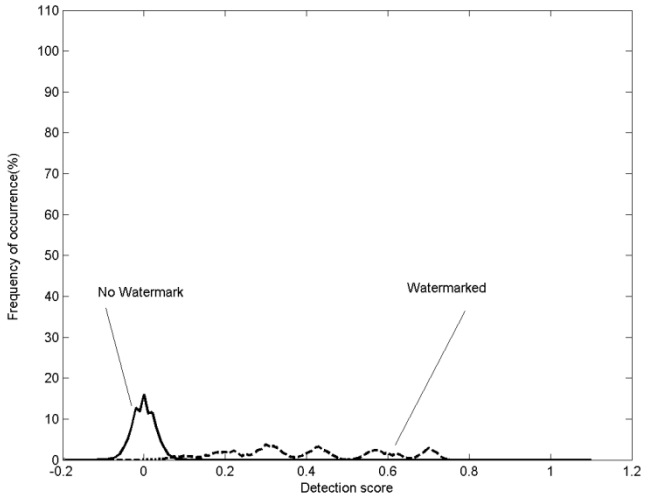

Fig. 25. Distribution of S_1 of MWE under LPF attack.

In attacks such as cropping, vector quantization, rotation, scaling, SWE may not be robust.

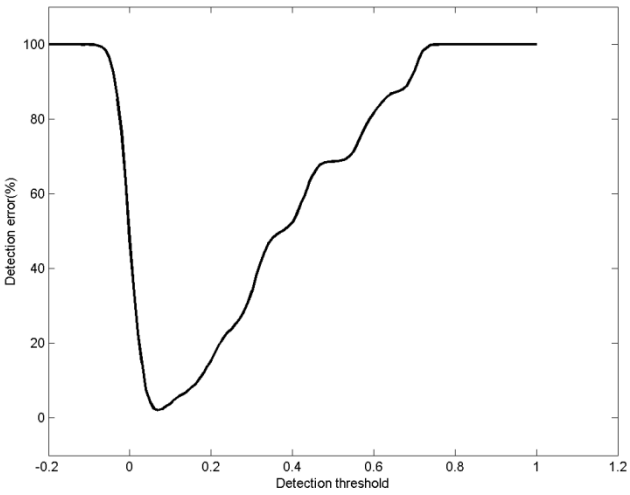

Fig. 26. Detection error of MWE under LPF attack.

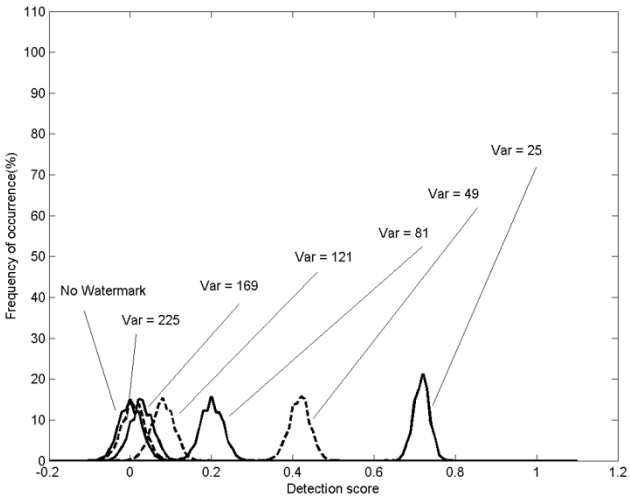

Fig. 27. Distribution of S_1 of MWE under noise attack.

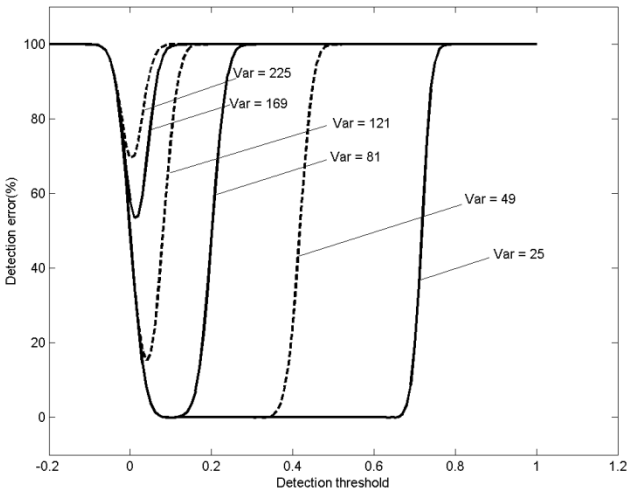

Fig. 28. Detection error of MWE under noise attack.

B. Results of MWE

The proposed DA, IA-R and IA-F for MWE are used to embed the "UST" logo in Lena. The average PSNR of the wa-

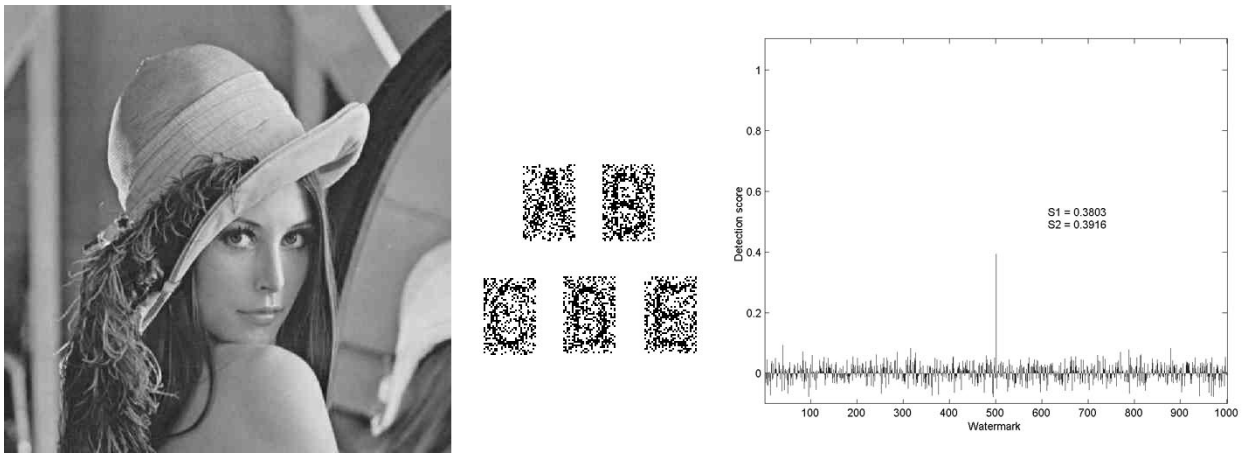

Fig. 29. (Left) MWE-watermarked image under JPEG attack, $\mathrm{SF} = 2, \mathrm{PSNR} = 33.46$ dB, $\mathrm{bpp} = 0.412$. (Middle) Decoded watermark. (Right) Random watermark detection results.

termarked images are shown in Fig. 19 against Q, the number of simultaneously embedded watermark bit sequences. All the values are averaged over ten trials with different random keys. The simulation is done with both orthogonal random subvectors $K_{j,i}$ and nonorthogonal subvectors. The curves marked DA, IA-R, and IA-F correspond to the nonorthogonal cases. The curve marked ORV corresponds to DA in the orthogonal case. In the orthogonal case, both IA-F and IA-R degenerate into DA because DA is the optimal solution. The keys $\boldsymbol{D_1}, \boldsymbol{D_2}, \ldots, \boldsymbol{D_Q}$ are generated from a Gaussian distribution with a mean of 250 and a variance of 4. The other keys $\boldsymbol{K_1}, \boldsymbol{K_2}, \ldots, \boldsymbol{K_Q}$ are generated from a Gaussian distribution with a mean of 0 and a variance of 16. As in SWE, these values are chosen in an ad-hoc way to achieve a PSNR of about 45 dB for watermarked images when $\mathrm{Q} = 5$.

In Fig. 19, regardless of the algorithm, the watermarked images have very high PSNR and very good visual quality when only one $(Q = 1)$ watermark is embedded. When more watermarks are embedded simultaneously, the PSNR of the watermarked images decrease with Q. In the nonorthogonal cases, the PSNR of IA-R and IA-F are similar, both being significantly higher than that of the low-complexity DA. This verifies that DA is nonoptimal in the nonorthogonal cases, and both the IA-R and IA-F can improve over DA significantly. The results also verify that it is possible to achieve higher PSNR in the nonorthogonal cases (by IA-R and IA-F) than in the orthogonal cases.

Here is a comparison of IA-F and IA-R in terms of PSNR, complexity and robustness. Theoretically, IA-F should always have higher PSNR than IA-R, but Fig. 19 suggests that their PSNR difference is insignificant. Their complexity are shown in Fig. 20 in terms of the average number of distinct vectors in the valid candidate vector pool. They have similar complexity for up to five watermarks $(\mathrm{Q} = 5)$, beyond which IA-F becomes significantly more complex than IA-R. In the Matlab 6.5 implementation on a Pentium 4 1.4-GHz PC, the average CPU time for IA-R and IA-F with $\mathrm{Q} = 5$ is 17.11 and 59.61 s, respectively.

As expected, all the watermark bits can be decoded perfectly under no attack resulting in both detection scores S_1 and S_2 being 1. The average detection score S_1 of MWE with five watermarks embedded $(\mathrm{Q} = 5)$ under JPEG-compression attack is shown in Fig. 21 against the JPEG scaling factor (SF). The S_1 of IA-F, IA-R, and DA are very similar at any SF. Considering the PSNR, complexity and performance under attacks, it appears that IA-R gives better quality-complexity-robustness tradeoff than IA-F.

Here are more robustness results of IA-R. The five binary logos in Fig. 6(b)–(f) are embedded with MWE using IA-R. A typical MWE-watermarked image (IA-R with $\mathrm{Q} = 5$) is shown in Fig. 22. With PSNR of 44.94 dB, the IA-R image has very good visual quality. The sample distributions of S_1 of MWE (IA-R, $\mathrm{Q} = 5$) watermark detection are shown in Figs. 23, 25, and 27 for JPEG attack, LPF, and noise attack, respectively. The corresponding total detection errors (sum of type 1 and 2 error probability) against different detection thresholds are shown in Figs. 24, 26, and 28.

In the JPEG attack on MWE in Fig. 23, unlike the case of SWE, the distribution of "no watermark" intersects with the distribution corresponding to $\mathrm{SF} = 3$. As a result, no threshold can give zero detection error probability at $\mathrm{SF} = 3$ in Fig. 24. A typical example of MWE under JPEG attack ($\mathrm{SF} = 2$, zero error) is shown in Fig. 29. Similar observations can be made for the LPF attack on MWE in Figs. 25, 26, and 30. With five watermarks embedded, the sample distributions of "watermarked" and "no watermark" are intersecting slightly. Error can occur in some cases. An example of MWE with no error is shown in Fig. 30. In the noise attack on MWE in Fig. 27, the distribution of "no watermark" intersects with many distributions and thus no zero detection error is possible in the corresponding situations in Fig. 28. Fig. 31 is an example of MWE when no error occurs. Fig. 32 is an example of the MWE under print-and-scan attack. Comparing these with the SWE results, it appears that more embedded watermarks tend to result in lower robustness.

C. Results for IWE in JPEG-Compressed Domain

The proposed IWE is simulated for the special situation when the original image is a JPEG image and the watermarked image is JPEG-compressed (with same parameters as original) to form a JPEG image. IWE is simulated with similar setup as in SWE to embed the "UST" logo. The results are shown in Figs. 33–35.

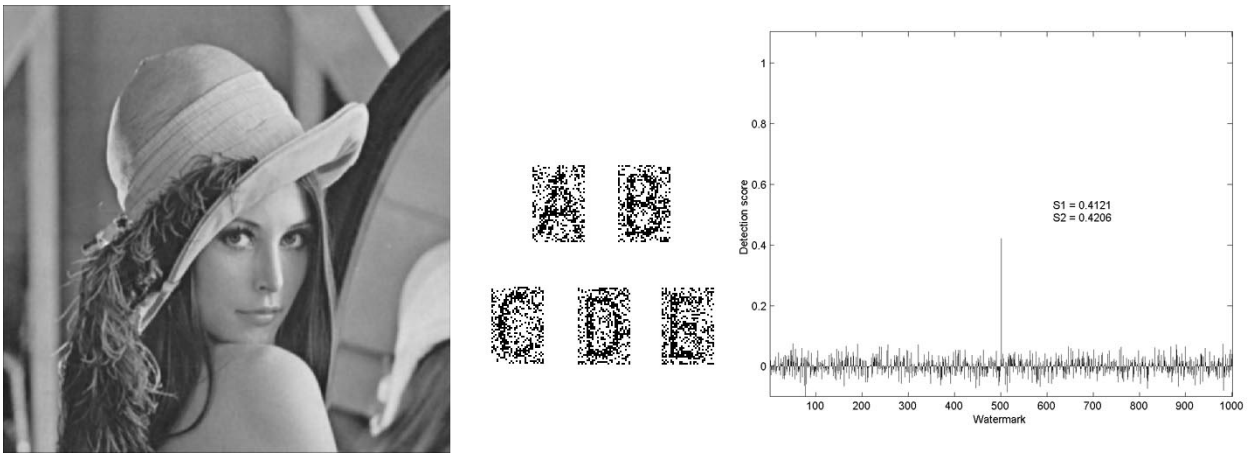

Fig. 30. (Left) MWE-watermarked image under LPF attack. $\mathrm{PSNR} = 31.82$ dB. (Middle) Decoded watermark. (Right) Random watermark detection results.

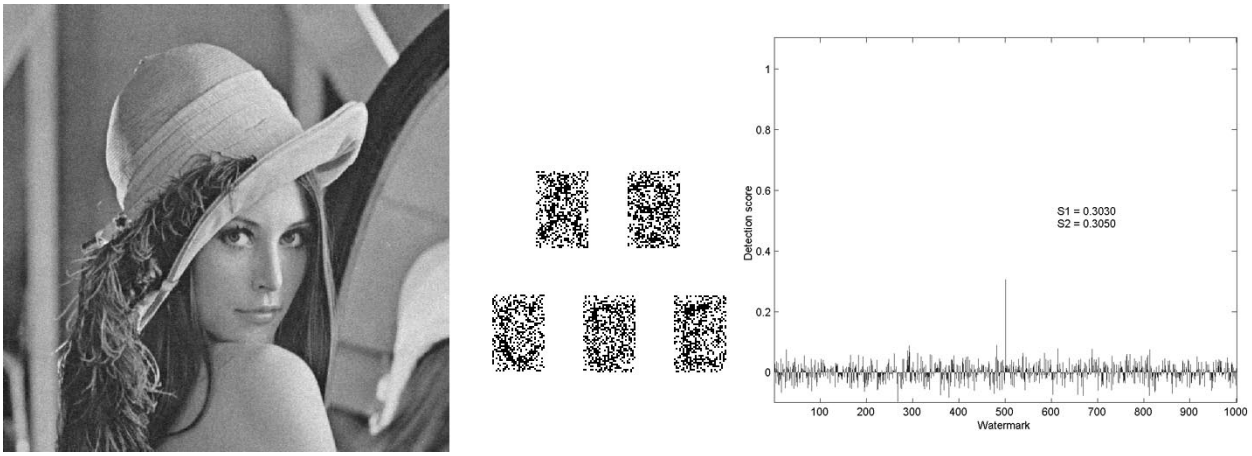

Fig. 31. (Left) MWE-watermarked image under noise attack. $\mathrm{PSNR} = 29.94$ dB, $\mathrm{variance} = 64$. (Middle) Decoded watermark. (Right) Random watermark detection results.

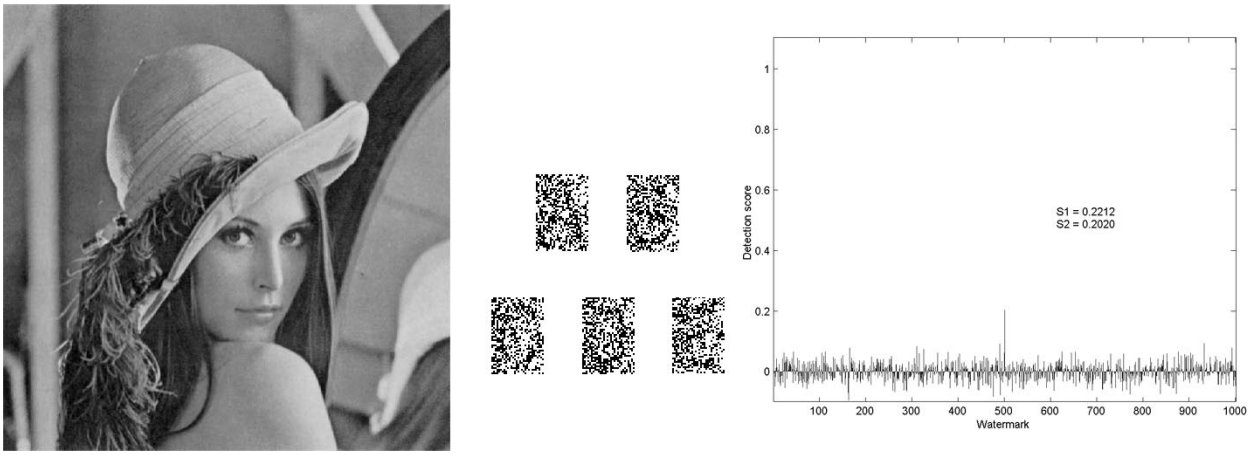

Fig. 32. (Left) MWE-watermarked image under print-and-scan attack. $\mathrm{PSNR} = 27.41$ dB. (Middle) Decoded watermark. (Right) Random watermark detection results.

Comparing Figs. 8 and 33, the proposed IWE can achieve significantly higher detection scores than SWE alone. In Fig. 33, S_1 of IWE starts to be less than one (bit errors start to occur) at SF greater than 1. With only limited number of iterations (1000 in our experiments) allowed, IWE may be unable embed a watermark bit in a bad situation. With increasingly severe compression, there are growing amount of such bad situations. Comparing the PSNR of "watermarked" and "no watermark", the visual quality of IWE remains good, with a drop of about 0.5 dB in PSNR. Two typical images are found in Figs. 42 and 43, showing the images before and after IWE. Fig. 35 shows that the complexity of IWE can be very large when SF is large.

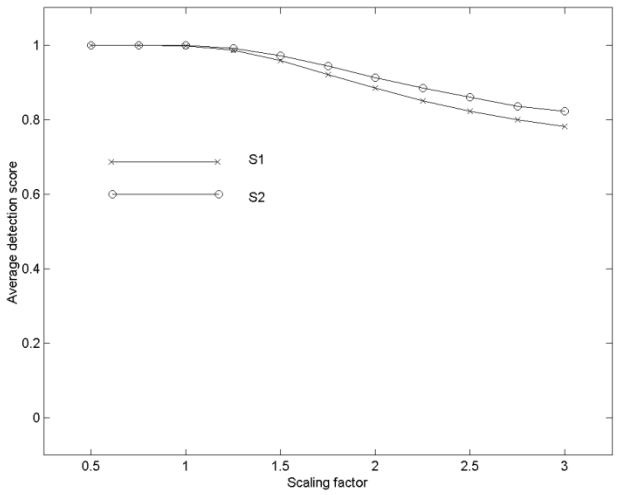

Fig. 33. Average detection score of IWE for JPEG images versus SF of original JPEG images (no attack).

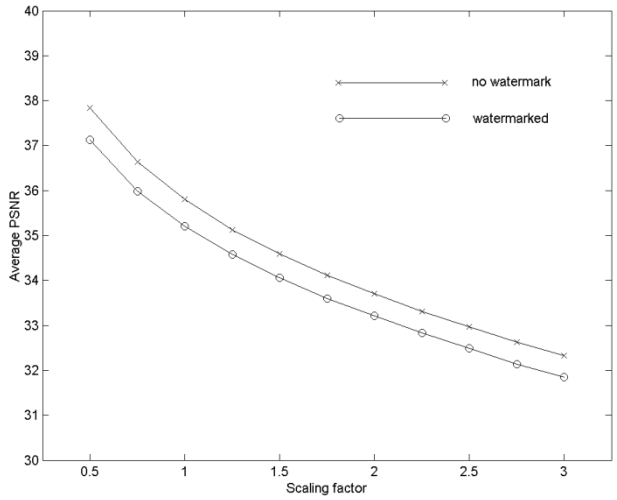

Fig. 34. Average PSNR of IWE-watermarked images versus SF of original JPEG image (no attack).

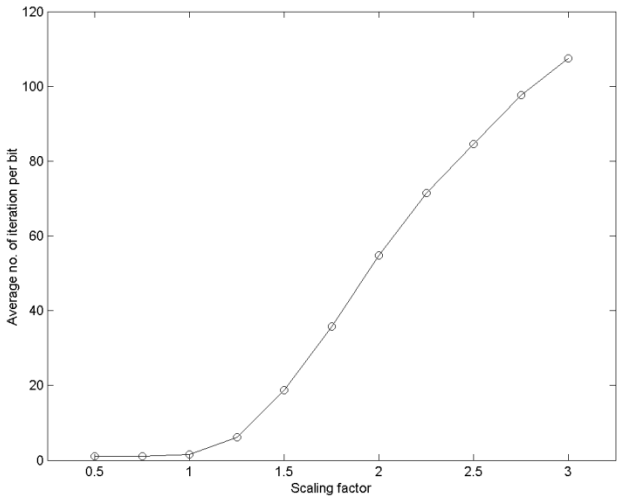

Fig. 35. Average number of iterations of IWE per embedded bit.

The distributions of S_1 of IWE under JPEG, LPF, and noise attacks are shown in Figs. 36, 38, and 40, respectively. The

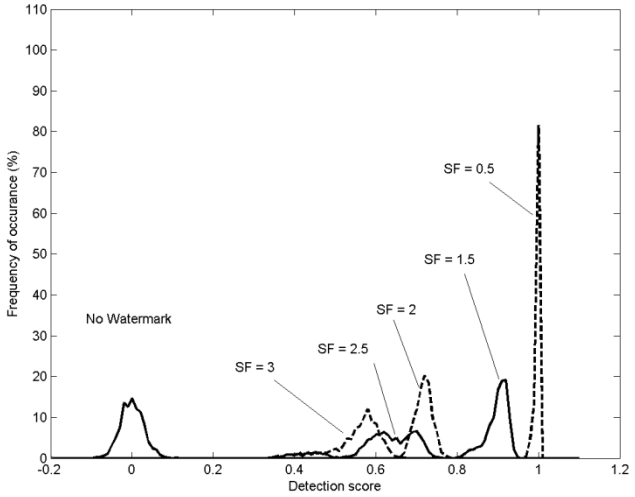

Fig. 36. Distribution of S_1 of IWE under JPEG attack.

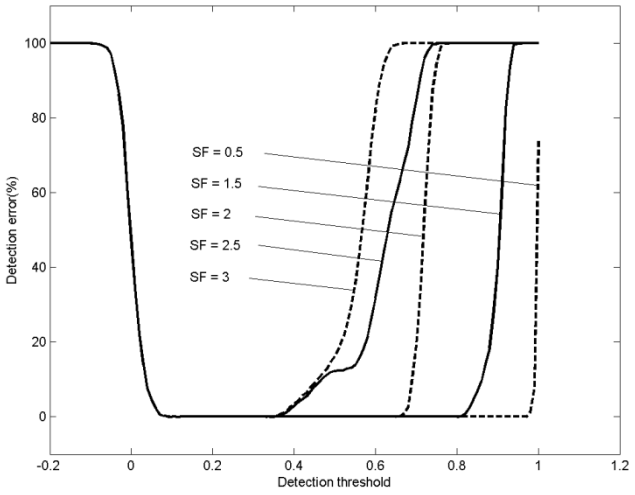

Fig. 37. Detection error of IWE under JPEG attack.

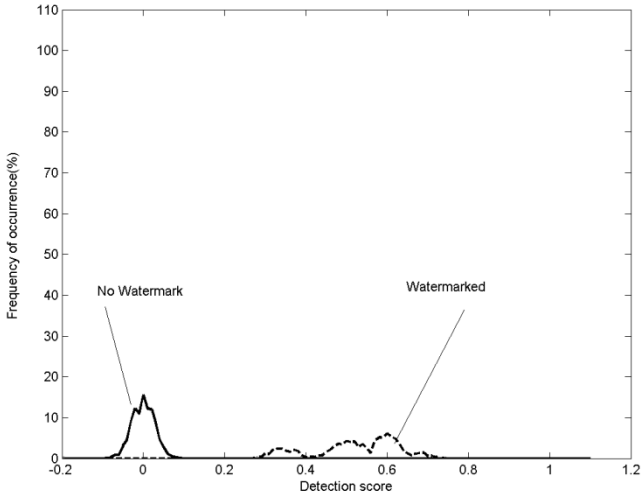

Fig. 38. Distribution of S_1 of IWE under LPF attack.

corresponding total detection errors against different detection thresholds are shown in Figs. 37, 39, and 41. In these simulations, the original image is a JPEG image compressed at $\mathrm{SF} = 1$ (as shown in Fig. 42) and the watermarked image using IWE is

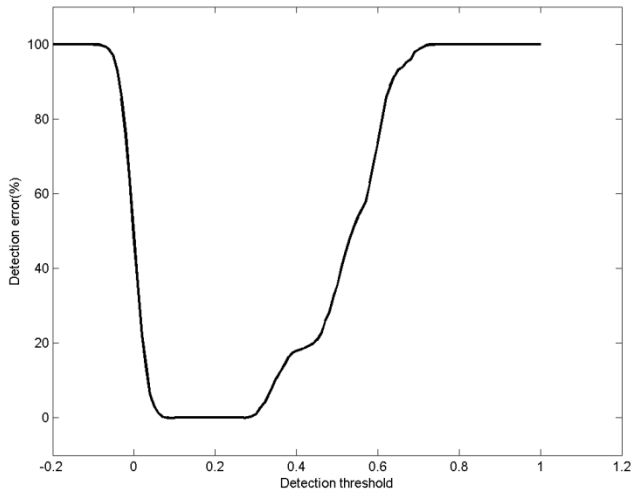

Fig. 39. Detection error of IWE under LPF attack.

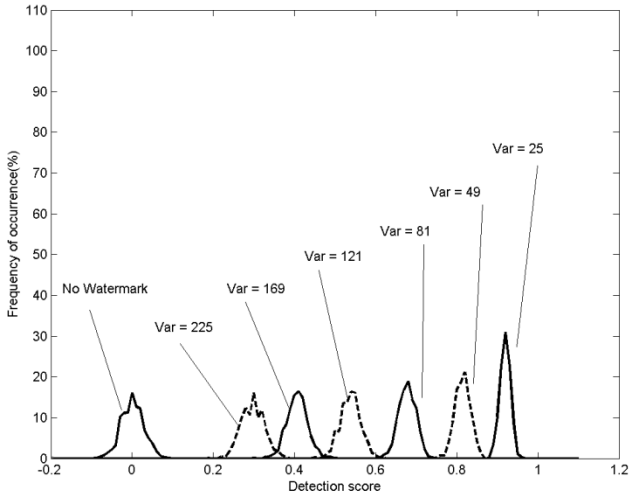

Fig. 40. Distribution of S_1 of IWE under noise attack.

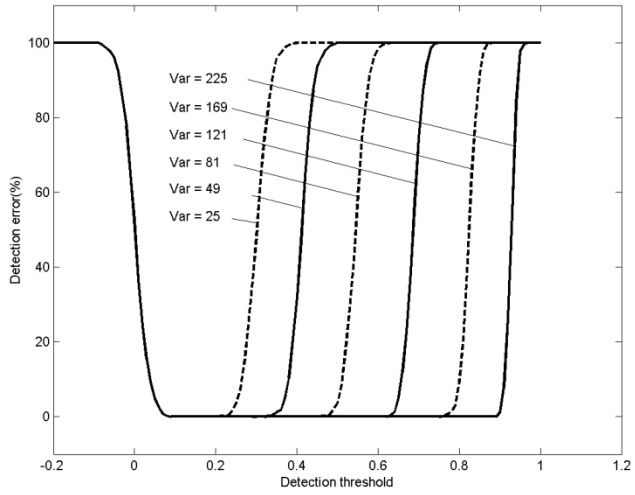

Fig. 41. Detection error of IWE under noise attack.

also a JPEG image compressed at $\mathrm{SF} = 1$ (as shown in Fig. 43). JPEG attack is now a transcoding attack or JPEG recompression

Fig. 42. Original JPEG image. $\mathrm{SF} = 1, \mathrm{PSNR} = 35.81\ \mathrm{dB}, \mathrm{bpp} = 0.642$.

Fig. 43. IWE-watermarked JPEG image. $\mathrm{SF} = 1$, $\mathrm{PSNR} = 35.18$ dB, $\mathrm{bpp} = 0.626$.

at a different SF. In the three attacks, the sample distribution of "no watermark" does not intersect with the other distributions and thus zero detection error can be achieved. Although the decoded logo may be noisy, the watermark is clearly distinguishable in the random watermark detection test. Typical examples of the three attacks and the print-and-scan attacks are shown in Figs. 44–47.

VI. Conclusion

In this paper, we propose three novel blind watermarking schemes to embed watermarks into digital images. The watermarks are designed to be decoded or detected without the original images. The SWE can embed one watermark bit sequence. The MWE can use correlated keys to embed multiple watermark bit sequences simultaneously such that individual watermark bit sequence can be decoded or detected independently. The IWE can embed watermark in a JPEG file and ensure it is detectable. Experimental results show that the three proposed watermarking algorithms give watermarked images with good visual quality. The embedded watermark is robust in varying degrees to unintentional attacks such as JPEG compression, transcoding, LPF, additive noise, and print-and-scan.

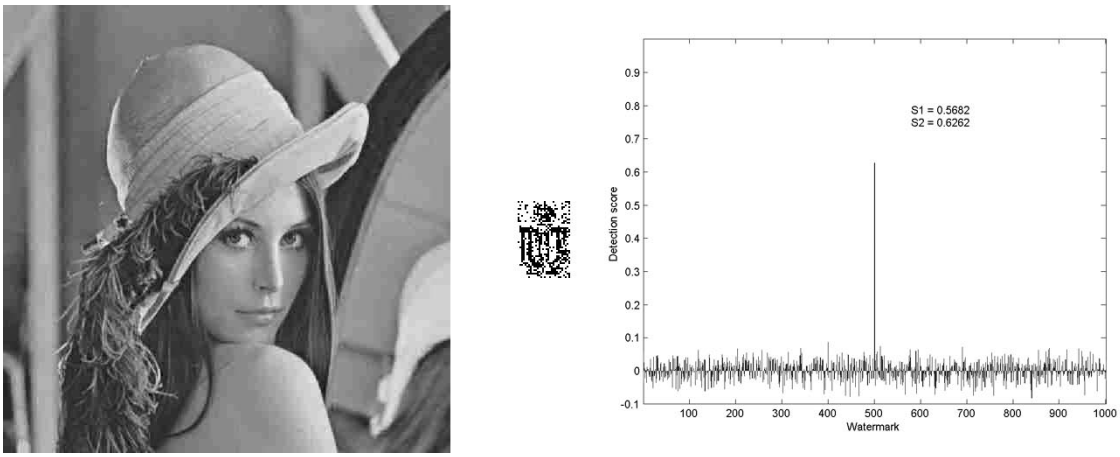

Fig. 44. (Left) IWE-watermarked image under JPEG transcoding attack. $\mathrm{SF} = 3$. $\mathrm{PSNR} = 32.07$ dB, $\mathrm{bpp} = 0.3245$. (Middle) Decoded watermark. (Right) Random watermark detection results.

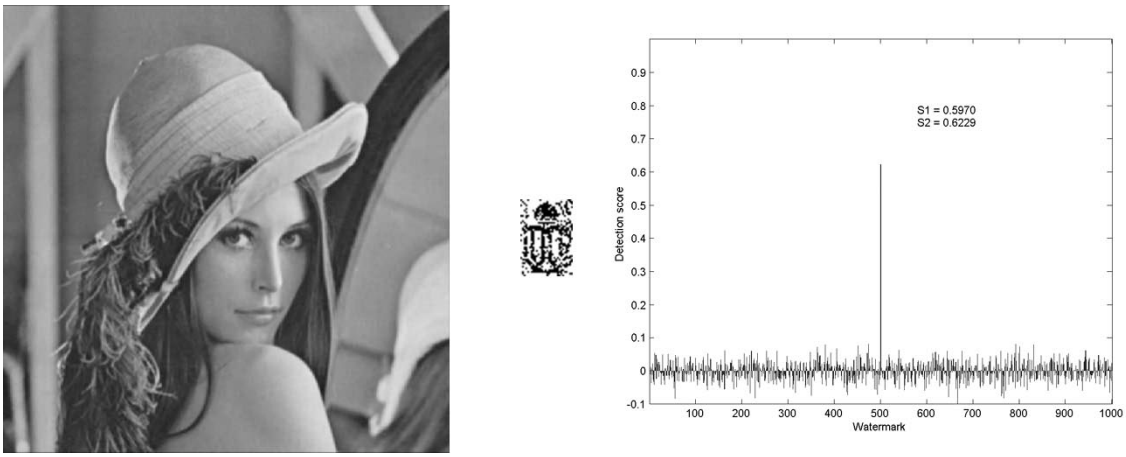

Fig. 45. (Left) IWE-watermarked image under LPF attack. $\mathrm{PSNR} = 31.64$ dB. (Middle) Decoded watermark. (Right) Random watermark detection results.

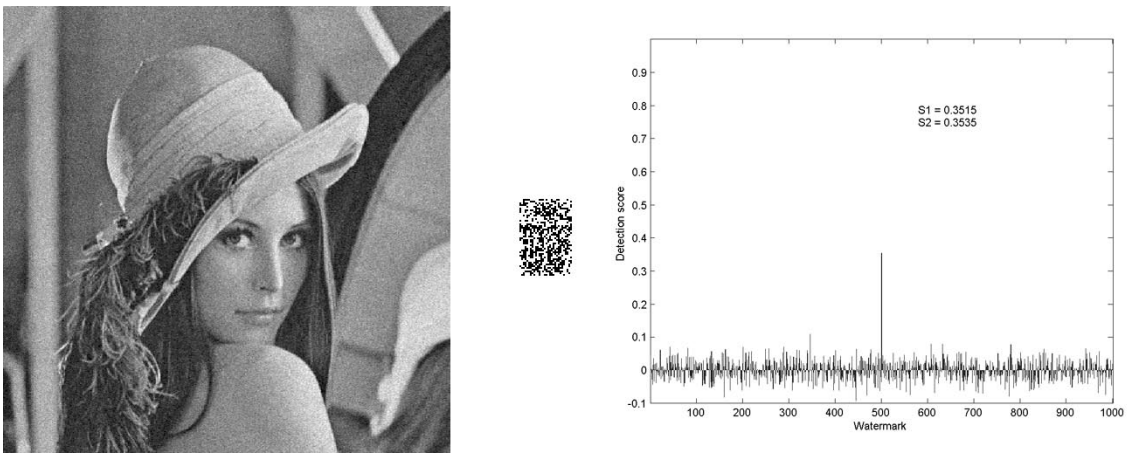

Fig. 46. (Left) IWE-watermarked image under noise attack. $\mathrm{PSNR} = 24.24$. $\mathrm{variance} = 225$. (Middle) Decoded watermark. (Right) Random watermark detection results.

APPENDIX

Consider $S_2 = \sum_{i=1}^{N} b_i(2 \cdot w_1 - 1) \cdot (2 \cdot w_i{}' - 1)$ where $b_i = (\beta_i^2 / \sum_{j=1}^{N} \beta_j^2)$ such that $b_i \geq 0$ and $\sum_{i=1}^{N} b_i = 1$. Let $X_i = 2w_i - 1$ and $Y_i = 2w_i{}' - 1$ such that both X_i and Y_i would take on values -1 and $+1$. Assuming that both w_i and w_i' are equally likely to be '0' or '1' such that the expected values

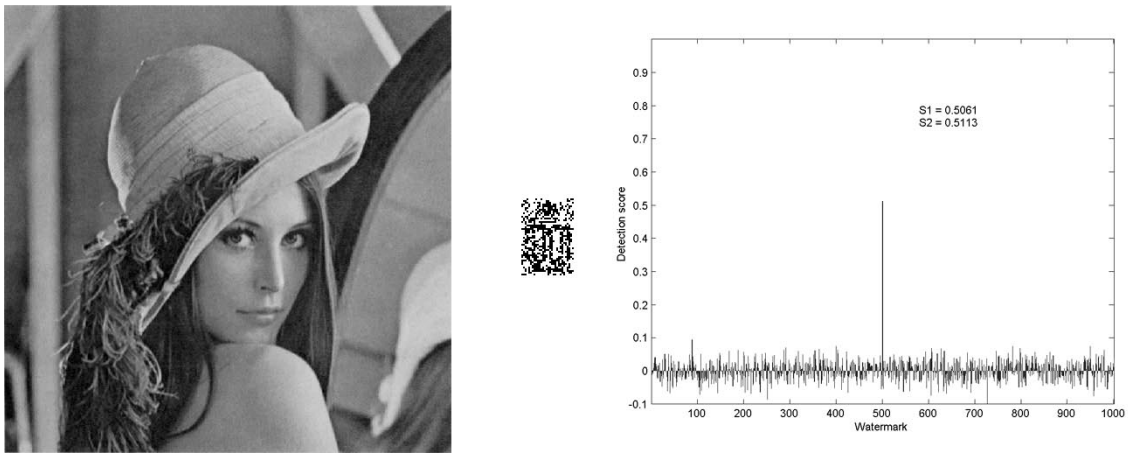

Fig. 47. (Left) IWE-watermarked image under print-and-scan attack. $\mathrm{PSNR} = 24.39$ dB. (Middle) Decoded watermark. (Right) Random watermark detection results.

of both X_i and Y_i are zero. When w_i and w_i' are independent such that X_i and Y_i are independent, we have

$$\begin{aligned} E(S_i) &= E\left(\frac{1}{N}\sum_{i=1}^{N} X_i Y_i\right) = \frac{1}{N}\sum_{i=1}^{N} E(X_i Y_i) \\ &= \frac{1}{N}\sum_{i=1}^{N} E(X_i)E(Y_i) = 0 \\ E(S_2) &= E\left(\sum_{i=1}^{N} b_i X_i Y_i\right) = \sum_{i=1}^{N} b_i E(X_i Y_i) \\ &= \sum_{i=1}^{N} b_i E(X_i)E(Y_i) = 0. \end{aligned}$$

When there is no noise/attack, $w_i = w_i'$ and $X_i = Y_i$ with probability 1 and

$$\begin{aligned} E(S_1) &= \frac{1}{N}\sum_{i=1}^{N} E(X_i Y_i) = \frac{1}{N}\sum_{i=1}^{N} E\left({X_i}^2\right) = 1 \\ E(S_2) &= \sum_{i=1}^{N} b_i E(X_i Y_i) = \sum_{i=1}^{N} b_i E\left({X_i}^2\right) = \sum_{i=1}^{N} b_i = 1 \end{aligned}$$

When there is noise/attack, $P(X_i = Y_i) < 1$. Suppose that the process is stationary such that $P(X_i = Y_i) = p < 1$ for all i. Then

$$\begin{aligned} E(S_1) &= \frac{1}{N}\sum_{i=1}^{N} E(X_i Y_i) \\ &= \frac{1}{N}\sum_{i=1}^{N} [1 \bullet P(X_i = Y_i) - 1 \bullet P(X_i \neq Y_i)] \\ &= p - (1-p) = 2p - 1 \\ E(S_2) &= \sum_{i=1}^{N} b_i E(X_i Y_i) \\ &= \sum_{i=1}^{N} b_i \left[1 \bullet P(X_i = Y_i) - 1 \bullet P(X_i \neq Y_i)\right] \\ &= [p - (1-p)]\sum_{i=1}^{N} b_i = 2p - 1 \end{aligned}$$

Without loss of generality, suppose the probability of error is equal for the first N_1 bits and separately equal for the other bits. In other words, $P(X_i = Y_i) = p_1 < 1$ for $1 \leq i \leq N_1$, and $P(X_i = Y_i) = p_2 < 1$ for $N_1 + 1 \leq i \leq N$ with $p_1 < p_2$. Let $N_2 = N - N_1$. If we choose b_i such that $b_i = B_1$ for $1 \leq i \leq N_1$, and $b_1 = B_2$ for $N_1 + 1 \leq i \leq N$ with $B_2 > (1/N) > B_1$, then

$$\begin{aligned} E(S_1) &= \frac{1}{N}\sum_{i=1}^{N} [1 \bullet P(X_i = Y_i) - 1 \bullet P(X_i \neq Y_i)] \\ &= \frac{N_1}{N}(2p_1 - 1) + \frac{N_2}{N}(2p_2 - 1) \\ &= \frac{N_1}{N}(2p_1) + \frac{N_2}{N}(2p_2) - 1 \\ E(S_2) &= \sum_{i=1}^{N} b_i \left[1 \bullet P(X_i = Y_i) - 1 \bullet P(X_i \neq Y_i)\right] \\ &= N_1 B_1(2p_1 - 1) + N_2 B_2(2p_2 - 1) \\ &= N_1 B_1(2p_1) + N_2 B_2(2p_2) - 1. \end{aligned}$$

Let $\Delta = p_2 - p_1 > 0$. Then

$$\begin{aligned} E(S_1) &= \frac{N_1}{N}(2p_1) + \frac{N_2}{N}(2p_1 + 2\Delta) - 1 \\ &= 2p_1 - 1 + \frac{2\Delta N_2}{N} \\ E(S_2) &= N_1 B_1(2p_1) + N_2 B_2(2p_2) - 1 \\ &= N_1 B_1(2p_1) + N_2 B_2(2p_1 + 2\Delta) - 1 \\ &= 2p_1 - 1 + 2N_2 B_2 \Delta. \end{aligned}$$

Therefore, $E(S_2) - E(S_1) = 2N_2\Delta(B_2 - (1/N)) > 0$.

ACKNOWLEDGMENT

The authors thank the reviewers for their valuable comments and suggestions on this paper.

REFERENCES

[1] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Processing*, vol. 6, pp. 1673–1687, Dec. 1997.

[2] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*. New York: Morgan Kaufmann, 2002.

[3] M. Kutter and F. A. P. Petitcolas, "A fair benchmark for image watermarking systems," in *Proc. Security and Watermarking of Multimedia Contents*, Jan. 1999, pp. 226–239.
[4] C. S. Lu, S. K. Huang, C. J. Sze, and H. Y. M. Liao, "Cocktail watermarking for digital image protection," *IEEE Trans. Multimedia*, vol. 2, pp. 209–224, Dec. 2000.
[5] C. T. Hsu and J. L. Wu, "Multiresolution watermarking for digital images," *IEEE Trans. Circuits Syst. II*, vol. 45, pp. 206–216, Aug. 1998.
[6] ——, "Hidden digital watermarks in images," *IEEE Trans. Image Processing*, vol. 8, pp. 55–68, Jan. 1999.
[7] J. Huang, Y. Q. Shi, and Y. Shi, "Embedding image watermarks in DC components," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 974–979, Sept. 2000.
[8] F. Hartung, *Digital Watermarking and Fingerprinting of Uncompressed and Compressed Video*. **AUTHOR: PLS. SUPPLY CITY**, Germany: Shaker Verlag, 2000.
[9] P. H. W. Wong, O. C. Au, and J. W. C. Wong, "Image watermarking using spread spectrum technique in log-2-spatio domain," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 1, May 2000, pp. 224–227.
[10] C. S. Lu and H. Y. M. Liao, "Multipurpose watermarking for image authentication and protection," *IEEE Trans. Image Processing*, vol. 10, pp. 1579–1592, Oct. 2001.
[11] J. R. Hernandez, F. P. Gonzalez, J. M. Rodriguez, and G. Nieto, "Performance analysis of a 2-D-multipulse amplitude modulation scheme for data hiding and watermarking of still images," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 510–524, May 1998.
[12] J. R. Hernandez, M. Amado, and F. Perez-Gonzalez, "DCT-domain watermarking techniques for still images: detector performance analysis and a new structure," *IEEE Trans. Image Processing*, vol. 9, pp. 55–68, Jan. 2000.
[13] G. C. Langelaar and R. Lagendijk, "Optimal differential energy watermarking of DCT encoded images and video," *IEEE Trans. Image Processing*, vol. 10, pp. 148–158, Jan. 2001.
[14] P. W. Wong and N. Memon, "Secret and public key image watermarking schemes for image authentication and ownership verification," *IEEE Trans. Image Processing*, vol. 10, pp. 1593–1601, Oct. 2001.
[15] Y. J. Zhang, T. Chen, and J. Li, "Embedding watermarks into both DC and AC components of DCT," in *Proc. SPIE Security and Watermarking of Multimedia Contents III*, Jan. 2001, pp. 424–435.
[16] C. Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, M. L. Miller, and Y. M. Lui, "Rotation, scale, and translation resilient watermarking for images," *IEEE Trans. Image Processing*, vol. 10, pp. 767–782, May 2001.
[17] V. Solachidis and I. Pitas, "Circularly symmetric watermark embedding in 2-D DFT domain," *IEEE Trans. Image Processing*, vol. 10, pp. 1741–1753, Nov. 2001.
[18] V. Licks and R. Jordan, "On digital image watermarking robust to geometric transformation," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, Sept. 2000, pp. 690–693.
[19] S. Stankovic, I. Djurovic, and I. Pitas, "Watermarking in the space/spatial-frequency domain using two-dimensional Radon-Wigner distribution," *IEEE Trans. Image Processing*, vol. 10, pp. 650–658, Apr. 2001.
[20] Y. Choi and K. Aizawa, "Digital watermarking using inter-block correlation: extension to JPEG coded domain," in *Proc. IEEE Int. Conf. Information Technology: Coding and Computing*, Mar. 2000, pp. 133–138.
[21] W. Luo, G. L. Heileman, and C. E. Pizano, "Fast and robust watermarking of JPEG files," in *Proc. IEEE 5th Southwest Symp. Image Analysis and Interpretation*, April 2002, pp. 158–162.
[22] S. Arena, M. Caramma, and R. Lancini, "Digital watermarking applied to MPEG-2 coded video sequences exploiting space and frequency masking," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, Sept. 2000, pp. 438–441.
[23] W. N. Lie, G. S. Lin, C. L. Wu, and T. C. Wang, "Robust image watermarking on the DCT domain," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 1, May 2000, pp. 228–231.

Peter H. W. Wong (M'01) received the B.Eng. degree (with first-class honors) in computer engineering from the City University of Hong Kong in 1996 and the M. Phil. degree in electrical and electronic engineering in 1998 from Hong Kong University of Science and Technology, Clear Water Bay, where he is currently working toward the Ph.D. degree.

His research interests include time-scale modification, motion estimation, and digital watermarking.

Mr. Wong received the Schmidt Award of Excellence in 1998.

Oscar C. Au (S'87–M'90–SM'01) received the B.A.Sc. degree from the University of Toronto, Toronto, ON, Canada, in 1986, and the M.A. and Ph.D. degrees from Princeton University, Princeton, NJ, in 1988 and 1991, respectively.

After a postion as a postdoctoral Researcher with Princeton Univ. for one year, he joined the Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology (HKUST), Clear Water Bay, in 1992, where he is now an Associate Professor and Director of the Computer Engineering Program. His main research contributions include video and image coding, watermarking and steganography, speech and audio processing. His topics of research are fast motion estimation for MPEG-1/2/4 and H.261/3/L, fast rate control, transcoding, post-processing, JPEG/JPEG2000, and halftone image data hiding, etc. His fast motion estimation algorithm was recently accepted by the ISO/IEC 14 496-7 MPEG-4 Standard. He is currently applying for five patents on his signal processing techniques. He has published over 100 technical journal and conference papers.

Dr. Au is an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS–I: FUNDAMENTAL THEORY AND APPLICATIONS. He is a member of the Technical Committee on Multimedia Systems and Applications and the Technical Committee on Video Signal Processing and Communications of the IEEE Circuits and Systems Society. He served on the organizing committee of the IEEE International Symposium on Circuits and Systems in 1997 and the IEEE International Conference on Acoustics, Speech and Signal Processing in 2003.

Y. M. Yeung received the B.Eng. degree (with first-class honors) in electrical and electronic engineering in 2001 from the Hong Kong University of Science and Technology, Clear Water Bay, where he is currently working toward the M.Phil. degree. His research interests include image/video coding, low-complexity coding algorithms, and multimedia coding standards