Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2008, Article ID 317242, 9 pages
doi:10.1155/2008/317242

Research Article

WMicaD: A New Digital Watermarking Technique Using Independent Component Analysis

Thang Viet Nguyen, Jagdish Chandra Patra, and Pramod Kumar Meher

School of Computer Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798

Correspondence should be addressed to Jagdish Chandra Patra, aspatra@ntu.edu.sg

Received 24 July 2006; Revised 22 February 2007; Accepted 15 August 2007

Recommended by B. Sankur

This paper proposes a new two-mark watermarking scheme that is based on the independent component analysis (ICA) technique. The first watermark is used for ownership verification while the second one is used as the copy ID of the image. Using a small-sized support image, the extraction is carried out on size-reduced level, bringing computational advantage to our method. The new method, undergoing a variety of experiments, has shown its robustness against attacks and its capability of detecting tampered area in the image.

1. INTRODUCTION

Digital watermarking, in which some information called the watermark is embedded directly and imperceptibly into original data (the so-called work), is one of the effective techniques to protect digital works from piracy [1, 2]. Once embedded, the watermark is bound to the work and should be extractable to prove the ownership, even if the work is modified [3]. Besides, it is preferable if the watermark also contains the tracking information about the copies of the work, that is, the copy ID. Because of its importance in digital media, watermarking has been extensively studied in recent years, with many approaches such as Fourier transform, Wavelet transform, QIM (quantization index modulation), and ICA (independent component analysis).

The idea of applying ICA to watermarking has been introduced in several studies, such as in the works of Zhang and Rajan [4], Gonzalez et al. [5], Bounkong et al. [6], and some others [7–9]. The similarity between ICA and watermarking schemes and the blind separation ability of ICA are the reasons that make ICA an attractive approach for watermarking.

In this contribution, we develop a novel method called WMicaD (watermarking by independent component analysis with dual watermark) that aims for the two above-mentioned goals: verifying the ownership and tracking the copies. To do it, the WMicaD method employs a dual watermark embedding scheme and an ICA-based extraction scheme. While the two watermarks allow us to verify the ownership as well as to track the copy ID, the ICA algorithm and watermark modification scheme allow us to extract the watermark with a single small-sized support image, the key image, without any information about the embedding parameters. Moreover, since the watermark extraction is carried out on size-reduced images, WMicaD gains computational advantage. In summary, our proposed method has the following characteristics.

(i) The size of the key image is much smaller than the original image. Thus, we need less storage memory space. Besides, the watermarked image may be made public if necessary.

(ii) The ICA-based extraction scheme does not require the original image and the watermarks. Also, the embedding parameters can be any arbitrary numbers.

(iii) The extraction is carried out on the down-sized images. It provides computational advantage compared to the extraction scheme with original size of the test image.

(iv) The proposed watermarking algorithm can serve for both ownership verification and image authentication.

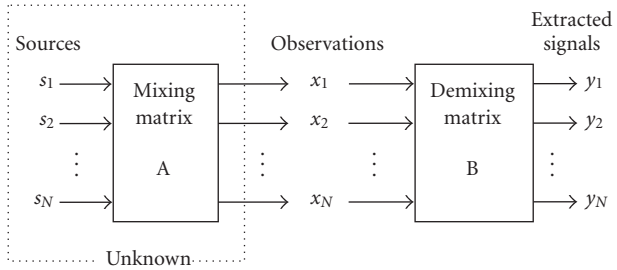


Figure 1: The ICA mixing and demixing models.

This paper is organized as follows. An overview of ICA and its similarity with watermarking is shown in Section 2. The WMicaD embedding and extraction schemes are detailed in Sections 3, 4, 5, and 6. We provide the computer simulations in Section 7. Finally, in Section 8, we conclude and discuss the issues related to the proposed algorithm.

2. WATERMARKING USING ICA

Independent component analysis (ICA) [10] is an important technique in signal processing whose goal is to unveil the hidden components from given observations. Assuming that the observed signals are mixtures of unknown independent sources, the ICA is carried out by finding a transform of the observation so that the new signals are as independent as possible [11]. Because of its blind extraction ability, many algorithms have been developed for ICA, for example, Infomax [12], FastICA [13], and ThinICA [14].

Shown in Figure 1 is the full ICA model which includes a mixing scheme and a demixing scheme. In the mixing scheme, the observed signals are generated by an unknown linear combination of the unknown sources. The scheme can be represented mathematically as

$$\mathbf{x} = \mathbf{A}s, \tag{1}$$

where $\mathbf{x} = [x_1, \ldots, x_N]^T$ is a vector of observed signals, $s = [s_1, \ldots, s_N]^T$ is a vector of original signals, and $\mathbf{A}_{N\times N}$ is a mixing matrix representing the unknown combination. This mixing scheme is similar to a watermark embedding scheme if we consider the work and the watermarks as unknown sources, and the watermarked images as the observations.

The goal of the ICA demixing scheme is to recover the hidden sources s_i, given the observations. It is similar to the watermark extraction scheme, where the watermarks are extracted from watermarked images. ICA carries out this task by maximizing the statistical independence criteria among the outputs $y_1, \ldots, y_N$ via a demixing matrix $\mathbf{B}$:

$$\mathbf{y} = \mathbf{B}\mathbf{x}. \tag{2}$$

When converged, $\mathbf{B}$ will be an inverse of $\mathbf{A}$ up to some permutations and scales, and $y_1, \ldots, y_N$ will be a permutation of the unknown sources $s_1, \ldots, s_N$. That is, if an ICA demixing scheme is applied on watermarked images, the outputs will be the embedded watermarks and the work.

Being interested in the potential of ICA, several authors have focused their studies on ICA-based watermarking [4, 5, 7–9]. As ICA algorithms require enough number of mixtures to run (the number of mixtures has to be equal to or more than the number of sources), a common challenge for ICA-based watermarking methods is to create different observations from the watermarked images and additional data. In [4, 5], the authors partitioned the original image into small blocks. The ICA algorithm was applied on these blocks to extract the independent components (ICs). Some of the less significant ICs were replaced by the watermarks. The watermarked image was then constructed from this new set of ICs. Major disadvantages of this approach, however, are the need of a large number of ICs and the high computational workload.

In [7], the authors used the original image and one of the two watermarks as the additional data. This is not preferable as the original image must be presented whenever ones want to proof the image ownership. In [9], the original image is not required but another watermarked image embedded by the same watermarks is needed. The extraction cannot be carried out without this large-size supporting image. Our proposed WMicaD method attempts to reduce the size of the supporting image by a watermark modification process. The modification is applied on the watermark so that it reveals different content on different image size.

3. WATERMARK MODIFICATION

In this paper, we treat a gray-level image, I of size $M \times N$, as a matrix of $M \times N$ whose entries are the pixel intensity values.

3.1. The downsizing and upsizing operators

The downsizing operator, denoted by $\mathcal{D}$, resizes an image of size $M \times N$ to k-time smaller images, $I_{[k](M/k)\times(N/k)} = \mathcal{D}(I_{M\times N}, k)$. The (m, n)th entry of the size-reduced image is the average of the pixel values inside a window of size $k \times k$ of the original image $I_{M\times N}$. That is,

$$I_{[k](m,n)} = \frac{1}{k^2}\sum_{i=0}^{k-1}\sum_{j=0}^{k-1} I_{(km+i,kn+j)}, \tag{3}$$

where k is a nonzero positive integer, called "resizing factor," $m = 0, 1, \ldots, (M - 1/k)$, and $n = 0, 1, \ldots, (N - 1/k)$.

The upsizing operator $\mathcal{U}$, in contrast, duplicates each element of $I_{M\times N}$ to every element in a window of size $k \times k$. The *k-time* upsized version of $I_{M\times N}$ is defined as $I^{[k]}_{kM\times kN} = \mathcal{U}(I_{M\times N}, k)$ whose (m, n)th entry is computed by

$$I^{[k]}_{(m,n)} = I_{(\lfloor m/k\rfloor,\lfloor n/k\rfloor)} \tag{4}$$

for all $m = 0, 1, \ldots, (kM - 1)$, $n = 0, 1, \ldots, (kN - 1)$, and k is the "resizing factor." The "floor" operator $\lfloor x\rfloor$ truncates the number x to the nearest smaller integer.

3.2. Watermark modification

As introduced in Section 2, we aim to embed the two watermarks (W_1 and W_2) into the original image. Hence, in order

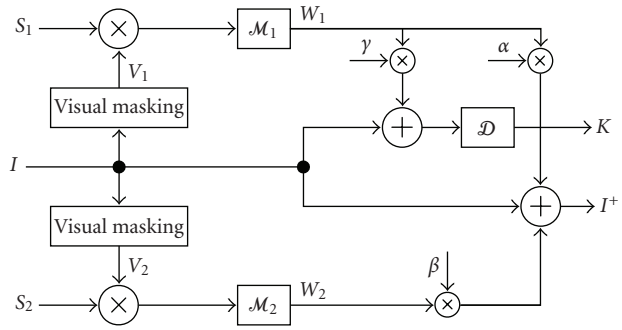


Figure 2: The WMicaD embedding scheme.

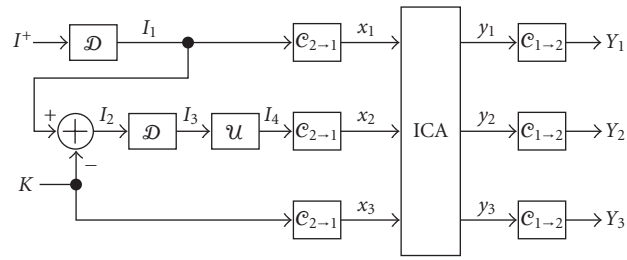


Figure 3: The WMicaD extraction scheme.

to apply ICA algorithm into the extraction scheme, we need at least three mixtures. However, we only have two available observations: a watermarked image and a small supporting image. Simple linear combination of these two images cannot create three independent mixtures. Therefore, our solution is to modify the watermarks with certain conditions so that they reveal different information at different image scales.

The first watermark, W_1, is modified in such a way that when it is downsized by a factor k_1, it produces a small-sized watermark, $W_{1[k_1]}$. But when W_1 is downsized by a factor k_1k_2, it produces a nullmatrix. Mathematically, this property can be expressed as

$$\varnothing_{[k_1k_2]} = \mathcal{D}(W_1, k_1k_2), \tag{5}$$

$$W_{1[k_1]} = \mathcal{D}(W_1, k_1), \tag{6}$$

where $\varnothing$ denotes a null matrix.

The second watermark, W_2, is modified so that when we downsize and subsequently upsize it again with the same factor, the watermark remains unchanged. Mathematically, this property can be expressed as

$$\mathcal{D}(W_2, k_1) = \mathcal{U}(\mathcal{D}(W_2, k_1k_2), k_2). \tag{7}$$

There are many ways to create the watermarks that satisfy (5), (6), and (7). In the appendices of this paper, we will introduce a simple modification method to create such watermarks. Also, in Section 5, we will explain in detail the use of the watermarks W_1 and W_2.

4. WMICAD EMBEDDING SCHEME

Shown in Figure 2 is the detail of our WMicaD embedding scheme. A watermarked image I^+ is generated by embedding two watermarks W_1 and W_2 into the original image, I. At the same time, a small-sized key image, K, is generated as the supporting image which will be used later in the watermark extraction.

We begin the embedding scheme by creating two visual masks V_1 and V_2 for the two watermarks. As discussed in [15], the visual masks help us to increase the embedding strength of the watermarks while maintaining the image's quality and watermark's invisibility. Our visual masks are computed from the original image, I, using NVF (noise visibility function) technique [15, 16].

Now, we create the watermarks from given signatures, S_1 and S_2. Visual mask V_1 and a modification function $\mathcal{M}_1$ (see the appendices) are applied on S_1 to generate the first watermark, W_1, that satisfies (5) and (6). Visual mask V_2 and modification function $\mathcal{M}_2$ are applied on S_2 to generate the second watermark, W_2, that satisfies (7).

In the last step, W_1 and W_2 are inserted into I to produce watermarked image I^+. Meanwhile, W_1 is combined with I and then downsized to produce the key image K. In summary, steps involved in the embedding scheme are given below.

(1) Create two visual masks V_1 and V_2 by NVF method. The visual mask V_1 can be different from V_2 by choosing different masking window half-lengths, $L_1 \neq L_2$.

(2) Create watermarks using modification functions

$$W_i = \mathcal{M}_i(S_i, V_i, k_1, k_2), \quad i = 1, 2, \tag{8}$$

where k_1, k_2 are the resizing factors.

(3) Create the watermarked image I^+ and the key image K:

$$I^+ = I + \alpha W_1 + \beta W_2, \tag{9}$$

$$K = \mathcal{D}(I + \gamma W_1, k_1). \tag{10}$$

Parameters α and β are called "embedding strengths" and γ is called "key-image coefficient." These parameters can be any nonzero values in the range of $[-1, 1]$.

5. WMICAD EXTRACTION SCHEME

Shown in Figure 3 is the detail of our WMicaD extraction scheme. We extract the two watermarks from the watermarked image, I^+, using ICA-based technique with support from the key image, K. As discussed earlier, firstly, we have to generate three mixtures and then apply ICA algorithm on them to receive the outputs. All of these processes will be carried out on size-reduced images.

The steps involved in the WMicaD extraction scheme are given below.

(1) Downsize the watermarked image I^+ to the size of the key image K with resizing factor k_1:

$$I_1 = \mathcal{D}(I^+, k_1). \tag{11}$$

(2) Create the image I_4 from I_1 and K by applying upsizing and downsizing operators with a resizing factor k_2,

$$I_2 = I_1 - K, \tag{12}$$
$$I_3 = \mathcal{D}(I_2, k_2), \tag{13}$$
$$I_4 = \mathcal{U}(I_3, k_2). \tag{14}$$

(3) Create 1D signals from I_1, I_4, and K,

$$[x_1, x_2, x_3]^T = [\mathcal{C}_{2\to1}(I_1), \mathcal{C}_{2\to1}(I_4), C_{2\to1}(K)]^T, \tag{15}$$

where $\mathcal{C}_{2\to1}$ denotes a 2D-to-1D operator.

(4) Apply an ICA technique on $\mathbf{x} = [x_1, x_2, x_3]^T$ to get three outputs $\mathbf{y} = [y_1, y_2, y_3]^T$.

(5) Convert back the outputs $\mathbf{y}$ to images,

$$Y_i = \mathcal{C}_{1\to2}(y_i), \tag{16}$$

where $i = 1, 2, 3$, and $\mathcal{C}_{1\to2}$ is a 1D-to-2D operator.

Now, let us see how the extraction scheme works on our special embedded watermarks. From (9) and (11), we have

$$I_1 = I_{[k_1]} + \alpha W_{1[k_1]} + \beta W_{2[k_1]}, \tag{17}$$

where $W_{1[k_1]}$ and $W_{2[k_1]}$ are the downsized images of W_1 and W_2 with resizing factor k_1. Similarly, we have

$$K = I_{[k_1]} + \gamma W_{1[k_1]}. \tag{18}$$

Replacing (17) and (18) into (12) and (13) yields

$$I_2 = (\alpha - \gamma) W_{1[k_1]} + \beta W_{2[k_1]}, \tag{19}$$
$$I_3 = (\alpha - \gamma)\mathcal{D}(W_{1[k_1]}, k_2) + \beta \mathcal{D}(W_{2[k_1]}, k_2). \tag{20}$$

Since W_1 satisfies (5) and (6), that is, $\mathcal{D}(W_{1[k_1]}, k_2) = \varnothing$, I_3 can be rewritten as

$$I_3 = \beta W_{2[k_1 k_2]}. \tag{21}$$

Finally, since W_2 satisfies (7), I_4 can be rewritten as

$$I_4 = \beta W_{2[k_1]}. \tag{22}$$

Using (17), (18), and (22), $\mathbf{x} = [x_1, x_2, x_3]^T$ can be represented as

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 & \alpha & \beta \\ 0 & 0 & \beta \\ 1 & \gamma & 0 \end{bmatrix} \begin{bmatrix} t_I \\ t_{W1} \\ t_{W2} \end{bmatrix}, \tag{23}$$

where t_I, t_{W1}, and t_{W2} are the three 1D signals, converted from $I_{[k_1]}$, $W_{1[k_1]}$, and $W_{2[k_1]}$ by the 2D-to-1D converters, respectively. That is, applying ICA algorithm on $\mathbf{x} = [x_1, x_2, x_3]^T$ results in the estimates of $I_{[k_1]}$, $W_{1[k_1]}$, and $W_{2[k_1]}$. And again, we can see that all the actions are taken on the downsized images, providing substantial computational advantage to WMicaD.

6. THE POSTPROCESSING SCHEME

As discussed in [11], one of the ambiguities of ICA is about the output order. In ICA, the outputs will be a permutation of the original sources. That is, we cannot say if the output y_1 corresponds to the source s_1, or whether y_2 is an estimate of s_2, and so on. Therefore, we develop a postprocessing scheme for our WMicaD method to identify the corresponding estimates, and to generate the estimates of the signatures from the estimated watermarks.

The postprocessing scheme is based on the correlation between each output Y_i, $i = 1, 2, 3$, and the watermarked image $I^+_{[k_1]}$ (in downsized version). To measure the similarity between two images, we use the absolute correlation coefficient (abCC). The absolute correlation coefficient between X and Y (both of size $M \times N$) is calculated by

$$|r_{X,Y}| = \frac{|s_{xy}|}{\sqrt{s_{xx} s_{yy}}}, \tag{24}$$

where

$$s_{xy} = \sum_{i=1}^{M}\sum_{j=1}^{N} (X_{(i,j)} - \overline{X})(Y_{(i,j)} - \overline{Y}),$$
$$s_{xx} = \sum_{i=1}^{M}\sum_{j=1}^{N} (X_{(i,j)} - \overline{X})^2,$$
$$s_{yy} = \sum_{i=1}^{M}\sum_{j=1}^{N} (X_{(i,j)} - \overline{Y})^2, \tag{25}$$
$$\overline{X} = \frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{N} X_{(i,j)},$$
$$\overline{Y} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} Y_{(i,j)}.$$

The abCC will approach 0 when two images are uncorrelated, and 1 when the two images are very similar to each other.

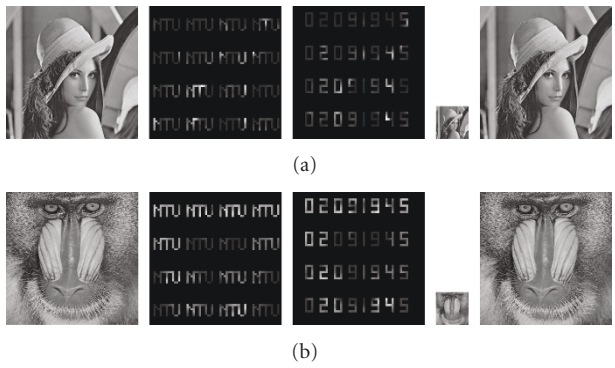
Now, we calculate the abCC between each output Y_i and $I^+_{[k_1]}$. Obviously, the output that corresponds to the original image will have a high abCC value. Whereas, the other outputs, which are the watermark's estimates, will have the abCC ≈ 0 since they are considered independent from the original image. Hence, taking the two outputs that have the lowest $|r_{I^+_{[k_1]}, Y_i}|$ will give us the estimates of the downsized watermarks $\widehat{W}_{1[K_1]}$ and $\widehat{W}_{2[K_1]}$.

In the next step, we obtain the original signatures from the watermark estimates. Since the watermarks are created by replicating the owner's signature, S_1, and the copy ID number, S_2, we partition the image Y_i into l subimages, $Y_{i1}, Y_{i2}, \ldots, Y_{il}$, each of size $M_S \times N_S$, where $M_S \times N_S$ is the size of the owner's signature. Averaging these subimages yields the estimate of the signature:

$$\hat{S} = \frac{1}{l}(\widehat{W}_{i1} + \widehat{W}_{i2} + \cdots + \widehat{W}_{il}). \tag{26}$$

(a) (b)

Figure 4: The two original signatures S_1 and S_2 used in the simulations. Both images are of size 16×64.



(a)

(b)

Figure 5: The images used in the WMicaD experiments. From left to right: original image I, watermarks W_1 and W_2, key image K, and watermarked image I^+. (a) Expt1: Lena image, (b) Expt2: Baboon image.

7. PERFORMANCE ANALYSIS

The robustness of the watermarked images was tested through various simulations under different attacks, including JPEG compression, gray-scale reduction, resizing, and noise addition. Besides, an authentication test was carried out to verify the WMicaD's ability of detecting the tampered area.

7.1. Simulation setup

Two binary images (16×64), a university name, and a copy ID, as shown in Figure 4, were used as the signatures during the embedding scheme. Two well-known gray-scale Lena and Baboon images, each of size 512×512, were used as the original images in the simulations. The original images, watermarks, watermarked images, and key images that were generated by WMicaD embedding scheme are shown in Figure 5.

In the embedding process, peak signal-to-noise ratio ($PSNR$) was chosen as the criterion to measure the quality of the watermarked image. The $PSNR$ between an image I and its modification $\hat{I}$ is defined as

$$PSNR = 20\log_{10}\left(\frac{255}{\sqrt{1/(MN)\sum_{i=1}^{M}\sum_{j=1}^{N}\left(X_{(i,j)} - \hat{X}_{(i,j)}\right)^2}}\right), \tag{27}$$

where $M \times N$ is the size of the two images. And for the extraction process, absolute correlation coefficient (abCC) ((24)) between the estimated signature and its original one, $|r_{\hat{S},S}|$, is chosen as the performance index.

To maintain the quality of the watermarked image and the imperceptibility of the watermarks, the embedding coefficients α, β and the window half-length L used in the visual mask function $\mathcal{V}$ were monitored so that $PSNR \geq 43dB$ in all experiments. The resizing factors k_1 and k_2 were also appropriately selected so that the key image K is small enough while the watermarks still have adequate details. Details of the parameters are provided in Table 1.

Table 1: The configuration table for the two experiments.

	α	β	γ	k_1	k_2	$L_1 = L_2$	$PSNR$
Lena	$\frac{-7}{256}$	$\frac{11}{256}$	$\frac{9}{256}$	4	2	12	44.99
Baboon	$\frac{-6}{256}$	$\frac{9}{256}$	$\frac{9}{256}$	4	2	10	43.14

With the chosen parameters, there is no noticeable difference between the original and watermarked images (see Figure 5). Moreover, the size of the key image (128×128) is 16 times reduced from the original 512×512.

In the next step, test images were generated by applying different attacks/modifications on the watermarked images. The WMicaD extraction and postprocessing scheme were carried out on the test images to estimate the signatures. The estimated signatures were then compared with the original ones, using abCC as the performance index to evaluate the quality of the estimation. In addition, we repeated the simulations with different ICA algorithms, such as SOBI (second-order blind identification) [17], JADETD (joint approximate diagonalization of eigen matrices with time delays) [18], and FPICA (fixed-point ICA) [13], in order to get a more general evaluation. It turned out that their results are almost identical. Thus, in this paper, we only show those simulations that were carried out with SOBI.

7.2. Common modification test

In this simulation, we tested the WMicaD method with three common image processing techniques: JPEG compression, gray-scale reduction, and resizing. A JPEG compression tool was used to compress the watermarked images with quality factor ranging from 90% down to 10%. In gray-scale reduction, the gray level was reduced from 256 down to $128, 64, \ldots, 8$ levels. And in resizing tests, the images were rescaled from 512×512 down to 128×128, and up to 1024×1024.

The results of WMicaD on the three tests are shown in Figure 6 and some illustrations of the estimated signatures are shown in Figure 7. In the two figures, "Expt1" and "Expt2" denote the performance plots of our experiments on the Lena and Baboon images, respectively. The symbols "−W1" and "−W2" represent the results on the first and second watermarks, respectively. As we can see, WMicaD produced good performance on all experiences. The quality of the estimates, in terms of abCC with the original signatures, is high even when the JPEG quality factor or the gray level is reduced to low value. Among the three modifications, simulations on resizing yielded the worst performance. It is probably due to the destruction of the first watermark's properties

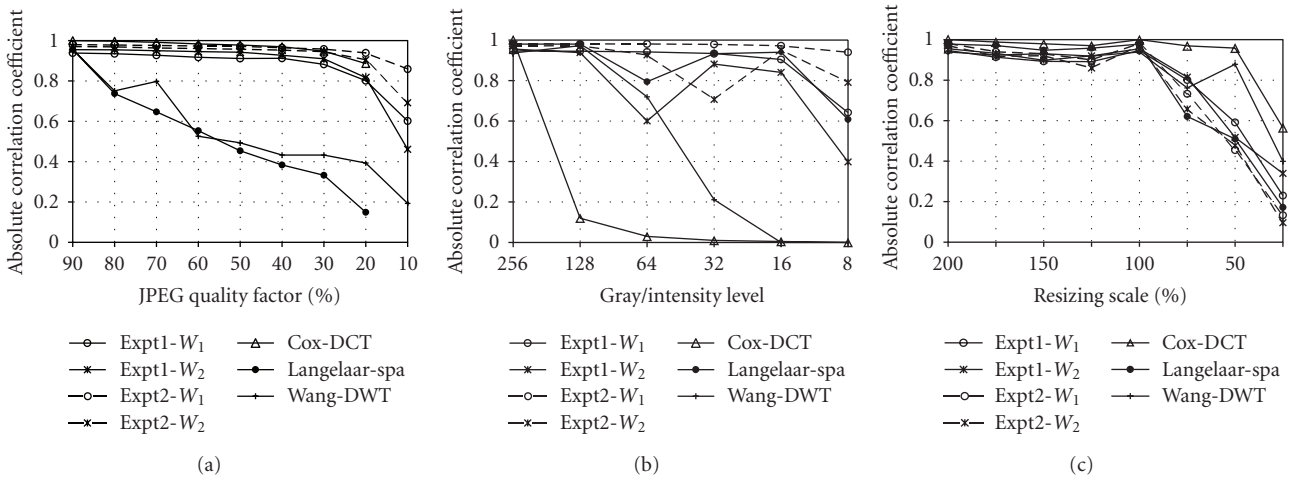


Figure 6: The performance results of WMicaD on three common attacks. (a) JPEG compression, (b) gray-level reduction, and (c) resizing. Expt1: Lena image; Expt2: Baboon image; W_1: first watermark; W_2: second watermark.

Figure 7: The estimated signatures by WMicaD extraction. From left to right: JPEG compression with quality factor of 50%, gray-level reduction down to 128, and resizing to the size of 384×384. (a) Experiment with Lena image and (b) Baboon image.

Table 2: Noise configuration table.

Gaussian	Mean $\mu = 0$, variance $\sigma^2 = [0 - 0.05]$
S&P	Noise density $[0 - 0.05]$
Multiplicative	Uniform noise Mean $\mu = 0$, variance $\sigma^2 = [0 - 0.05]$

(5) and (6) when the image is resized, that is, pixel values are interpolated.

For further investigation, we compared the proposed method with several well-known watermarking techniques that work on different processing domains [19]. These techniques include a discrete cosine transform algorithm *Cox-DCT* [20], a spatial domain algorithm *Langelaar-spa* [21], and a discrete wavelet transform algorithm *Wang-DWT* [22]. The Lena images (in Expt1) were used as the original images. Our copy ID signature (the number sequence) was chosen as the watermark. After the embedding process, the distortions of the watermarked images in terms of *PSNR* were found to be 38.4 dB , 34.2 dB, and 36.7 dB for the *Cox-DCT, Wang-DWT*, and *Langelaar-spa*, respectively. It may be noted that in our experiments, the *PSNR* is found to be 44.9 dB and 43.1 dB for Expt1 and Expt2 (see Table 1). The performance results of the watermark extraction were computed in term of the absolute correlation coefficient and they are shown in Figure 6. As it can be seen in Figure 6, WMicaD provided a competitive performance; it even yielded better results in JPEG and gray-level reduction tests. These are very encouraging results, considering that WMicaD uses two watermarks that are overlapped on each other.

7.3. Addition-of-noise test

From some points of view, an attack to the watermarked image can be considered as a noise being added to the image. Therefore, in this section, we investigate the performance of WMicaD under different types of noise, including Gaussian-noise, "salt and pepper" (S&P) noise, and multiplicative noise. Noise range and properties used in the simulations are presented in Table 2.

The simulation results of WMicaD on the noise tests are shown in Figure 8. The method provided good performance on the "S&P" noise and multiplicative noise experiments but not very impressive performance on Gaussian-noise test. This can be explained from the ICA property. As discussed in [11], in order to get a good ICA estimation, the source signals should be non-Gaussian. Therefore, when the Gaussian-noise was added, it made the sources more Gaussian and hence, a poor performance of the ICA-based extraction scheme.

More simulations on image rotation, cropping, brightness and contrast adjustments, and filtering have been carried out to measure the performance of WMicaD [16]. The method produces very good result on the brightness and contrast adjustment attacks. In the desynchronization attacks, such as rotation and cropping, WMicaD performance is not as good as on the JPEG compression test, but it is better than in the Gaussian-noise attack. For example, in rotation attack, we assumed that the rotation angle was unnoticeable to the extractor, that is, no preinverse rotation operation was applied. The extraction is carried out directly on the rotated image. The results were encouraging, and the estimated signatures are still recognizable even when the image was rotated by 0.25 degree.

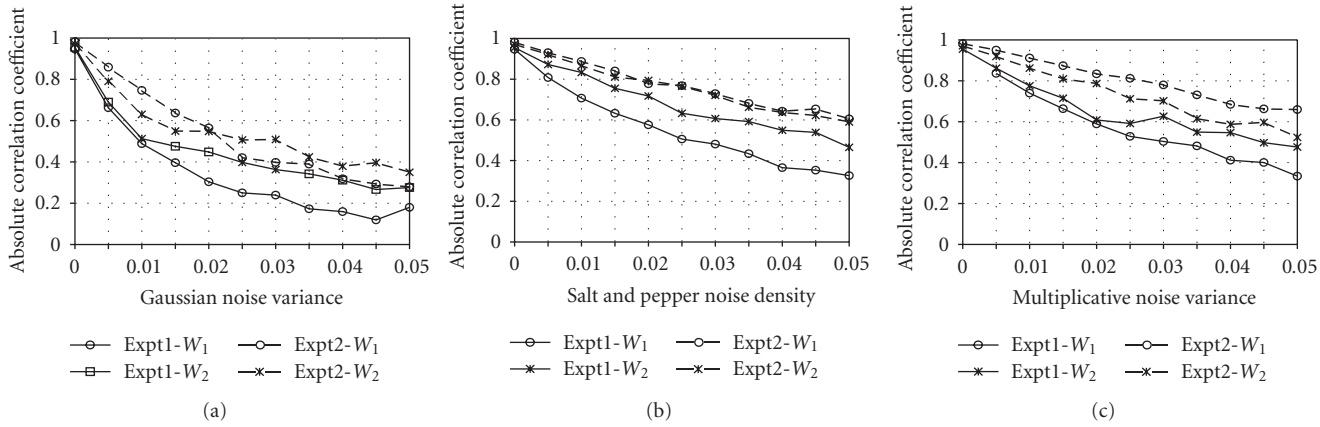


Figure 8: The performance results of WMicaD on noise tests. (a) Gaussian-noise, (b) S&P noise, and (c) multiplicative noise. Expt1: Lena image; Expt2: Baboon image; W_1: first watermark; W_2: second watermark.

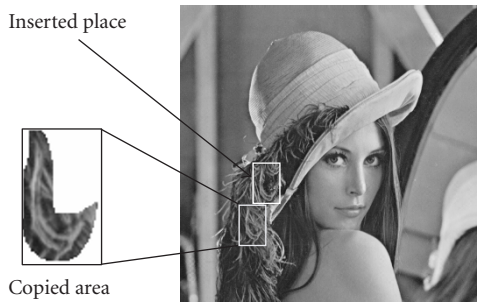


Figure 9: The image used in the tampering test. A small portion of the Lena image is copied and inserted in to another place (the tampering area is magnified and shown on the left side of the tampered image).

Figure 10: Three output images Y_1, Y_2, and Y_3 of WMicaD in the tampering test (all images are of size 128×128). The tampered area can be observed in the outputs Y_2 and Y_3 which correspond to the two watermark estimates.

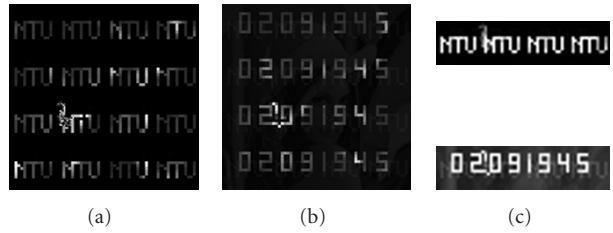


Figure 11: The estimated watermarks and signatures after the tampered area is corrected. (a) The first watermark, (b) the second watermark, and (c) the two estimated signatures.

7.4. WMicaD for detection of tampered area

The previous section has shown the ability of WMicaD in verifying the ownership. In this section, another ability of WMicaD in image authentication is introduced. The following experiment will demonstrate how WMicaD method is able to detect the tampered area in the image.

Shown in Figure 9 is Lena image that was tampered by a small portion of the image (the feather portion in the hat's tail area). This portion was copied and maliciously overwritten to another similar place in order to make it undetectable by naked eyes.

7.4.1. Detecting the tampered area

Now, we carry out the extraction scheme and carefully observe the three output images Y_1, Y_2, and Y_3. As it is shown in Figure 10, the tampered area, even if small, is clearly noticeable in the watermark estimates, with the pixel values of the tampered area being much higher than the rest of the images.

7.4.2. Recovering the signatures

After successfully detecting the tampered area, WMicaD is still able to extract the signature from the tampered image by doing an additional step before carrying out the postprocessing scheme. Here, we replace the pixel values in the tampered area (the area where pixel values are significantly high) by the average values of the other pixels (the pixels that are not inside the tampered area). Next, we quantize all the pixels of the image to 256 gray level. Finally, we put the corrected image to the postprocessing scheme to estimate the signatures. And as it is shown in Figure 11, the estimated watermarks and signatures are clearly visible and easy to recognize.

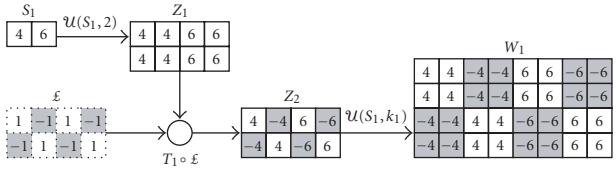


Figure 12: An example of the first watermark modification scheme $\mathcal{M}_1$. A watermark W_1 of size 8×4 is generated from an author signature S_1 of size 2×1. Resizing factor is $k_1 = k_2 = 2$.

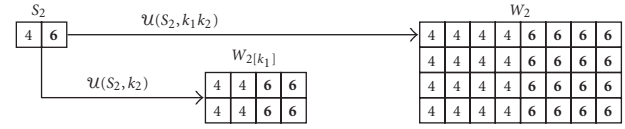


Figure 13: An example of the second watermark modification scheme $\mathcal{M}_2$. A watermark W_2 of size 8×4 is generated from an author signature S_2 of size 2×1. Resizing factor is $k_1 = k_2 = 2$.

8. DISCUSSION AND CONCLUSION

In this paper, we have proposed a novel watermarking method called WMicaD that embeds two watermarks into the host image. The unique two-watermark embedding scheme and the ICA-based extraction scheme have brought many interesting properties to WMicaD.

Firstly, this dual watermark embedding scheme allows us to achieve two goals at the same time: verifying the ownership of the image and tracking the copy ID of the original image. Unlike other watermarking algorithms that use a sequence of numbers as a single watermark, we apply images as the watermarks. Hence, at the extraction side, the estimated signatures can be easily verified by visual inspection. In addition, overlapping of watermarks makes them harder to be recognized in the host image.

Secondly, utilization of specially tailored watermarks and ICA algorithm in the extraction scheme makes it possible to estimate the watermarks without the original image, and without any information about the embedding parameters. Please note that while ICA is considered as a blind separation method, our WMicaD extraction is not considered as a totally blind watermarking extraction, since it uses a small supporting key image. We can embed the watermark with different embedding strengths (the alpha and beta parameters), and different copy IDs (the second watermark) on different image copies. Since all of the three parameters (alpha, beta, and gamma) can be changed in every image, it is almost impossible for the attackers to know these parameters. Thus, it helps to prevent the watermarks from being discovered or removed.

Theoretically, carrying out the extraction on size-reduced images brings to WMicaD a computational advantage. As seen in the simulations, the size of images was reduced by 4×4 times, resulting in a much more faster processing time in comparison with the extraction on the original images. Please note that if the other competitive algorithm also applies down-sizing operation before carrying out the watermark extraction, then our WMicaD might not have clear computational advantages. However, not every algorithm can carry out the extraction on the down-sized images. And even if it is possible, the quality of the estimated signatures is another topic that needs further investigation. In addition, size-reduced images also prevent the attackers from removing the watermarks from the host image, since the small-size estimated outputs are much different from the original one.

Through the simulations, we have used several watermarking algorithms for performance comparison using absolute correlation coefficient (abCC) as a performance index. It is good but not a perfect measure. Sometimes, an estimate with poor abCC is easy to observe than one with higher abCC. Also, since we are using a two-watermark embedding scheme and carrying out the extraction on size-reduced image, it is hard to have an absolute comparison. The comparison used in experiments should be considered as an illustration for our WMicaD performance. In addition, the performance is varied, depending on the content of the two watermarks as well as the original image.

APPENDICES

A. FIRST WATERMARK MODIFICATION

The goal of the first watermark modification function, $\mathcal{M}_1$, is to generate a watermark, W_1, from the owner's signature so that the watermark satisfies (5) and (6). Details of the scheme are provided in the following paragraphs and shown in Figure 12.

Let S_1 be an image of size $(M/k_1k_2) \times (N/k_1k_2)$ that represents the owner signature. The scheme to construct the watermark W_1 from the owner's signature is described by

$$W_1 = \mathcal{U}(\mathcal{U}(S_1, k_2) \bullet £, k_1). \tag{A.1}$$

First, the signature S_1 is upsized by a factor k_2 to create a matrix Z_1. Second, Z_1 is multiplied element by element with a "chessboard" matrix £ to produce Z_2. Finally, Z_2 is upsized by a factor k_1 to generate the watermark W_1. It can be seen that when W_1 is downsized by k_1k_2, it will result in a null matrix satisfying (5). In this scheme, the chessboard matrix £ is a matrix whose (m, n)th entry is defined by

$$£_{(m,n)} = \begin{cases} 1 & \text{if } (m+n) = \text{even}, \\ -1 & \text{otherwise}, \end{cases} \tag{A.2}$$

and the (m, n)th entry of the element-by-element product $\bullet$ is computed by

$$Z_{2(m,n)} = Z_{1(m,n)} £_{(m,n)}. \tag{A.3}$$

B. SECOND WATERMARK MODIFICATION

The second modification function $\mathcal{M}_2$ is to create a watermark W_2 that satisfies (7). Beginning with a signature S_2 of size $(M/k_1k_2) \times (N/k_1k_2)$, we apply the upsizing operator $\mathcal{U}$ on S_2 with resizing factors k_1k_2 to obtain

$$W_2 = \mathcal{U}(S_2, k_1k_2). \tag{B.1}$$

Shown in Figure 13 is an illustration of the second modification scheme, $\mathcal{M}_2$. The second watermark W_2 of size 8×4 is constructed from a signature S_2 of size 2×1 by an upsizing operator $\mathcal{U}$ with the resizing factors $k_1 = 2$ and $k_2 = 2$. It is easy to see that the generated watermark W_2 satisfies (7).

REFERENCES

[1] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*, edited by E. Fox, Morgan Kaufmann, San Francisco, Calif, USA, 1st edition, 2001.

[2] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1079–1107, 1999.

[3] P. Moulin and J. A. O'Sullivan, "Information-theoretic analysis of information hiding," *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 563–593, 2003.

[4] S. Zhang and P. K. Rajan, "Independent component analysis of digital image watermarking," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '02)*, vol. 3, pp. 217–220, Phoenix, Ariz, USA, May 2002.

[5] F. J. González-Serrano, H. Y. Molina-Bulla, and J. J. Murillo-Fuentes, "Independent component analysis applied to digital image watermarking," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01)*, vol. 3, pp. 1997–2000, Salt Lake, Utah, USA, May 2001.

[6] S. Bounkong, B. Toch, D. Saad, and D. Lowe, "ICA for watermarking digital images," *Journal of Machine Learning Research*, vol. 4, no. 7-8, pp. 1471–1498, 2003.

[7] M. Shen, X. Zhang, L. Sun, P. J. Beadle, and F. H. Y. Chan, "A method for digital image watermarking using ICA," in *Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA '03)*, pp. 209–214, Nara, Japan, April 2003.

[8] D. Yu, F. Sattar, and K.-K. Ma, "Watermark detection and extraction using independent component analysis method," *EURASIP Journal on Applied Signal Processing*, vol. 2002, no. 1, pp. 92–104, 2002.

[9] D. Yu. and F. Sattar, "A new blind watermarking technique based on independent component analysis," in *Digital Watermarking: 1st International Workshop (IWDW '03)*, vol. 2613 of *Lecture Notes in Computer Science*, pp. 37–73, Springer, Berlin, Germany, 2003.

[10] P. Comon, "Independent component analysis, a new concept?" *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.

[11] A. Cichocki and S.-I. Amari, *Adaptive Blind Signal and Image Processing*, John Wiley & Sons, New York, NY, USA, 2002.

[12] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, no. 6, pp. 1129–1159, 1995.

[13] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.

[14] S. A. Cruces and A. Cichocki, "Combining blind source extraction with joint approximate diagonalization: thin algorithms for ICA," in *Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA '03)*, pp. 463–468, Nara, Japan, April 2003.

[15] S. Voloshynovskiy, A. Herrigel, N. Baumgaertner, and T. Pun, "A stochastic approach to content adaptive digital imagewatermarking," in *Proceedings of the 3rd International Workshop on Information Hiding (IH '99)*, pp. 212–236, Dresden, Germany, September-October 1999.

[16] T. V. Nguyen, "Studies on independent component analysis for watermarking and nonlinear blind source separation," Ph.D. dissertation, School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore, May 2007.

[17] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Transactions on Signal Processing*, vol. 45, no. 2, pp. 434–444, 1997.

[18] P. Georgiev and A. Chichocki, "Robust independent component analysis via time-delayed cumulant functions," *IEICE Transactions on Fundamentals of Electronics*, vol. E86-A, no. 3, pp. 573–579, 2003.

[19] P. Meerwald, "Digital image watermarking in the wavelet transform domain," M.S. thesis, University of Salzburg, Salzburg, Austria, January 2001.

[20] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673–1687, 1997.

[21] G. C. Langelaar, J. C. A. van der Lubbe, and R. L. Lagendijk, "Robust labeling methods for copy protection of images," in *Storage and Retrieval for Image and Video Databases V*, vol. 3022 of *Proceedings of SPIE*, pp. 298–309, San Jose, Calif, USA, February 1997.

[22] H.-J. M. Wang, P.-C. Su, and C.-C. J. Kuo, "Wavelet-based digital image watermarking," *Optics Express*, vol. 3, no. 12, pp. 491–496, 1998.