Available online at www.sciencedirect.com
ScienceDirect
Neurocomputing 70 (2007) 2881–2890
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Independent component analysis in the blind watermarking of digital images

J.J. Murillo-Fuentes*

Dep. TSC. Escuela Sup. de Ingenieros. Paseo de los descubrimientos sn, 41092 Sevilla, Spain

Available online 22 May 2007

Abstract

We propose a new method for the blind robust watermarking of digital images based on independent component analysis (ICA). We apply ICA to compute some statistically independent transform coefficients where we embed the watermark. The main advantages of this approach are twofold. On the one hand, each user can define its own ICA-based transformation. These transformations behave as "private-keys" of the method. On the other hand, we will show that some of these transform coefficients have white noise-like spectral properties. We develop an orthogonal watermark to blindly detect it with a simple matched filter. We also address some relevant issues as the perceptual masking of the watermark and the estimation of the detection probability. Finally, some experiments have been included to illustrate the robustness of the method to common attacks and to compare its performance to other transform domain watermarking algorithms.

Keywords: Image processing; Watermarking; Copyright protection; Authentication; Independent component analysis

1. Introduction

We deal with blind robust watermarking (RW) of digital images. Image watermarking allows the protection of copyrighted data by ownership verification, preventing unauthorized distributions. Protection is achieved by embedding a piece of information, known as the watermark, within the to-be protected image. This original image is often regarded as the cover or host image. Usually, the embedded watermark must be imperceptible so that the quality of the document is not affected. In the robust approach, the main goal is to detect the embedded watermark even if it has been severely attacked, either intentionally or unintentionally. If we detect the watermark without using the cover image, the method is referred to as blind RW.

Direct embedding in the spatial domain is desirable for low-complexity requirements. However, it may be easily attacked. Other techniques based on transformed domains are usually more robust. In these methods, we insert the watermark into the transform coefficients. Transform domain embedding methods include the block-based discrete cosine transform (DCT), the discrete wavelet transform (DWT) and other frequency representations [2,10,11,23]. These algorithms usually have good properties against many attacks, including JPEG compression. However, as the transform domain used is usually known, we may perform an attack on the transform coefficients to remove the embedded watermark. Besides, these methods are not usually blind, as we need the original image to detect or retrieve the watermark from the watermarked image [10].

In this paper, we investigate the application of the independent component analysis (ICA) [8,9] to the blind RW. The basic idea of ICA RW is to project the image into independent components, the transform coefficients, to embed the watermark into them. In contrast to other methods [4,24,28] where the authors focus on the detection, we introduce great improvements in the embedding stage. ICA was first applied to the development of a blind robust embedding method for the watermarking of images in [20]. In the present paper, we deeply review this algorithm to greatly improve it. First, the method inherits the property

*Tel.: +34 954488150; fax: +34 954487341.
E-mail address: murillo@esi.us.es

0925-2312/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2006.06.011

of fulfilling one of the Kerckhoffs' [14] principles.[1] To cope with this, we propose to use different ICA projections as users' private keys. Second, in [20] we used an image as watermark and a simple correlator as detector. In this paper, we take advantage of the transformed coefficients presenting noise-like spectra. This property suggested us to use a spread spectrum (SS) watermark for the following reasons [11]. First of all, the watermark would present a similar spectrum to the transformed coefficients, making it more difficult to filter the watermark out. Also, since the SS watermark is independent (orthogonal) to the transformed coefficients, it can be efficiently blindly detected using a simple matched filter (MF). These characteristics and the resulting new algorithm have been developed in the paper as follows.

We introduce in Section 2 the ICA technique and its application to image processing. In Section 3, we overview the application of this tool to image watermarking. Section 4 is devoted to propose a new general ICA-based watermarking method in which we address both the embedding and the detection algorithms. In Section 5, we endow the method with practical features. We focus on the generation of the SS watermark, the insertion of the watermark using a perceptual masking, the extraction algorithm and the estimation of the probability of detection. In Section 6 we propose some attacks as experiments to illustrate the good performance of the method and to compare it to a DCT-based algorithm. We end the paper with some concluding remarks in Section 7.

2. ICA in image processing

2.1. Independent component analysis

ICA [9] consists of projecting a set of components onto another statistically independent set. This technique is usually applied to the blind separation of sources (BSS) [8], as statistical independence of the outputs ensures separation. These approaches assume a multiple-input multiple-output model and have been successfully applied to several subjects (see [8] and references therein) such as communications [7,19] biomedical signals like ECG or EEG [17], financial data or image processing [15,20]. In the simple BSS noiseless instantaneous linear model, the entries of a sample t of a column vector sequence $\boldsymbol{x}_t$ are l mixtures of l zero-mean independent signals, the sources $\boldsymbol{s}_t$. In matrix form, it follows that

$$\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{s}_t, \tag{1}$$

where $\boldsymbol{A}$ is the mixing matrix. We then apply ICA to project $\boldsymbol{x}_t$ into a space of l components $\boldsymbol{y}_t$ as statistically independent as possible. This projection is represented by an $l \times l$ separating matrix $\boldsymbol{B}$:

$$\boldsymbol{y}_t = \boldsymbol{B}\boldsymbol{x}_t = \boldsymbol{B}\boldsymbol{A}\boldsymbol{s}_t = \boldsymbol{U}\boldsymbol{s}_t. \tag{2}$$

If $\boldsymbol{A}$ is non-singular, $\boldsymbol{x}_t$ is stationary and at most one of the entries of $\boldsymbol{s}_t$ is Gaussian distributed then the independent components $\boldsymbol{y}_t$ are an estimate of the original sources [9,25] up to scaling and permutations. In this sense, a matrix $\boldsymbol{U}$ is non-mixing if it has just one non-zero entry per column and per row.

Very much literature has been devoted to ICA algorithms. We will use here batch algorithms that minimize the marginal entropies (ME) of the outputs [9]. These algorithms have a good performance and are easier to use than gradient-based methods [1,19] as they do not need any parameter definition. Assuming zero-mean unit-variance uncorrelated components, the minimization of the marginal entropies of the outputs may be approximated [9] by the minimization of

$$\phi^o_{\mathrm{ME}}(\boldsymbol{y}) \approx -\frac{1}{48}\sum_i (C^y_{iiii})^2, \tag{3}$$

where the marginal cumulant of output i yields $C^y_{iiii} = \mathrm{E}[y_i^4] - 3$. Notice that this cost function only assumes that the sources have non-zero fourth-order cumulants. This is a quite minor restriction in practice. In the 2-dimensional case, we can easily minimize this cost function[2] [22]. Then, we face the n-dimensional case by decomposing it into 2-dimensional problems, i.e., by using the Jacobi optimization (JO) method [9].

2.2. ICA in image processing

There are two common applications of ICA to image processing. On the one hand, we may assume we have l linear mixtures of l images. Therefore, we simply need to reshape each mixture of images into a vector and then apply ICA to separate them as in Eq. (2). On the other hand, we may find more elaborated approaches where only one image is involved. These methods first decompose an image into components $\boldsymbol{x}_t$ to later apply ICA [3]. Afterwards, any image processing technique may be applied to these, so computed, independent components $\boldsymbol{y}_t$ [15]. Notice that each row of the separating matrix $\boldsymbol{B}$ in (2) provides one independent component (an entry of vector $\boldsymbol{y}_t$). Therefore, if we reshape each row of $\boldsymbol{B}$ into a matrix, we get a set of 2-dimensional basis functions. These basis functions, also regarded as patches or features, are closely related to well-localized and oriented Gabor filters [13]. Some other authors suggest these basis functions to be the edges of the image [3,12], or even to model the receptive fields of the primary visual cortical neurons [13]. An analysis of the image independent components shows that many of them are sparse distributed and that only some basis functions are needed to represent the image. Besides, the probability of independent components having small amplitudes is high, but large amplitudes occurs as well [13]. In [15], these features are used to compress or encode an

[1] The main Kerckhoffs' principle states that "The security of the encryption scheme must depend only on the secrecy of the key, and not on the secrecy of the algorithm."

[2] A loss function is usually referred to as 'contrast function' in the ICA literature.

image. Basic compression algorithms exploit these ideas as they retain only the independent components with larger energy. In addition, the authors in [15] show that groups of images with similar features may be restored from a common set of basis (rows of matrix $\boldsymbol{B}$). Hence, it is possible to use ICA to define a set of basis functions to encode a group of images such as natural images or text scans. This idea in [6,15] can be further extended to the use of the projection matrix $\boldsymbol{B}$ computed for one image in the processing of another one. Particularly, if they belong to the same class of images (text images, natural scenes,…).

In the following, we will describe how to rearrange an image $\boldsymbol{I}$ into its components $\boldsymbol{x}_t^I$ to later apply ICA to compute $\boldsymbol{y}_t^I$. We also include some practical examples to illustrate the main ideas above.

Assume matrix $\boldsymbol{I}$ be a gray-scale image of size $n \times m$ as in Fig. 1. This matrix can be divided into blocks to later reshape them into vectors $\boldsymbol{x}_t^I$. The entries of $\boldsymbol{x}_t^I$ are the components or mixtures to be projected into independent components. We describe this process in detail as follows. Matrix $\boldsymbol{I}$ is divided into $k \times k$ blocks $\boldsymbol{C}_{p,q}$ according to,

$$\boldsymbol{C}_{p,q}(i,j) = \boldsymbol{I}((p-1)\cdot k + i, (q-1)\cdot k + j), \tag{4}$$

where

$$\begin{aligned} &i,j = 1,2,\ldots,k;\\ &p = 1,2,\ldots,n/k;\\ &q = 1,2,\ldots,m/k. \end{aligned} \tag{5}$$

Then, matrix $\boldsymbol{C}_{p,q}$ is reshaped row wise into vector $\boldsymbol{x}_t^I$, where $t = (p-1)\cdot m/k + q$, following

$$x^I_{((p-1)\cdot m/k+q)}((i-1)\cdot k + j) = \boldsymbol{C}_{p,q}(i,j). \tag{6}$$

We will denote this transformation by $\Upsilon(\cdot)$, i.e.,

$$\boldsymbol{x}_t^I = \Upsilon(\boldsymbol{I}, k). \tag{7}$$

The rows of $\boldsymbol{x}_t$ may be then projected onto $l = k^2$ independent components,

$$\boldsymbol{y}_t^I = \boldsymbol{B}\boldsymbol{x}_t^I, \quad t = 1,\ldots,mn/k^2. \tag{8}$$

In Fig. 1 we include a sketch of this process for $k = 2$. We use this transformation of the image into components as we got samples from all around the image. But other transformations, e.g., randomly selected $k \times k$ patches, are also possible.

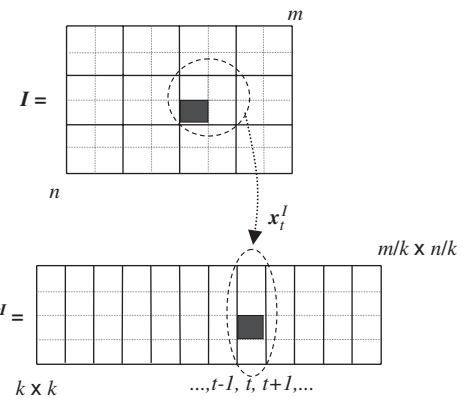

Fig. 1. Sketch of the image transformation process for $k = 2$.

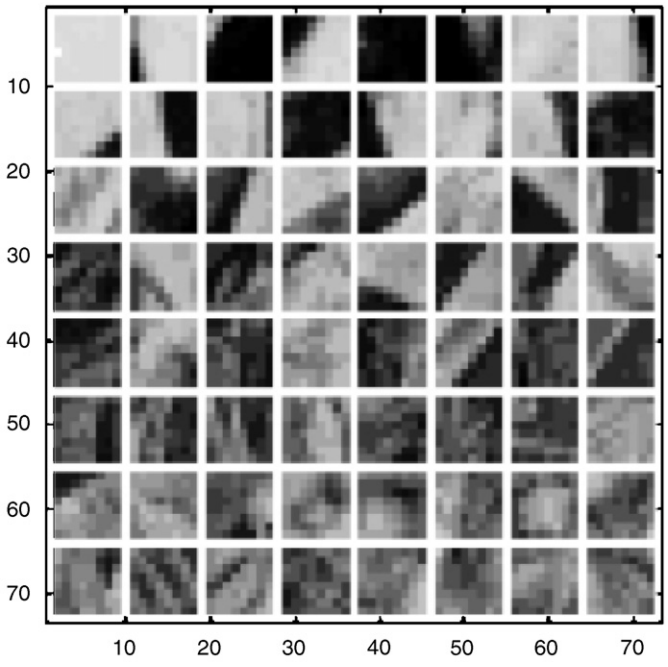

Fig. 2. Example of basis functions for $k = 8$.

In Fig. 2 we show the spatial basis functions computed for $k = 8$ and the Lena Image (256×256). Each row of the separating matrix $\boldsymbol{B}$ was reshaped into a 8×8 image. Then, these patches were arranged row wise in descending order of energy, i.e., those basis functions (rows of $\boldsymbol{B}$) providing independent components with larger variance are located at the top rows. The top-left corner basis represents the DC component of every 8×8 patch of the image. Notice also that the first rows are the basis functions to build the borders of the image. Besides, the last ones provide low energy components, details of the image. In Fig. 3 we depict the power spectral density (PSD) for the independent components number 1 (Fig. 3a) , 5 (Fig. 3b) and 9. (Fig. 3c) computed for the image of Lena with $k = 3$. As the independent components have been arranged in descending order of energy, we have the DC component in Fig. 3a. The independent component 5 in Fig. 3b has a white noise-like frequency response. Finally, the last independent component has a high frequency response due to the high sparsity of its values.

3. ICA in watermarking of digital images

The applications of ICA to image processing described above lead to different watermarking algorithms. The simplest approach is as follows [28]. The cover (or original) image $\boldsymbol{I}$ and the watermark $\boldsymbol{W}$ are reshaped into vectors, $\boldsymbol{x}_t^I$ and $\boldsymbol{x}_t^W$, and then the watermark is inserted into the cover image by mixing them. The simplest mixing is the addition of the watermark to the host image in the spatial domain $\boldsymbol{x}_t^I + \boldsymbol{x}_t^W$. The watermark can also be implemented in any transform domain such as

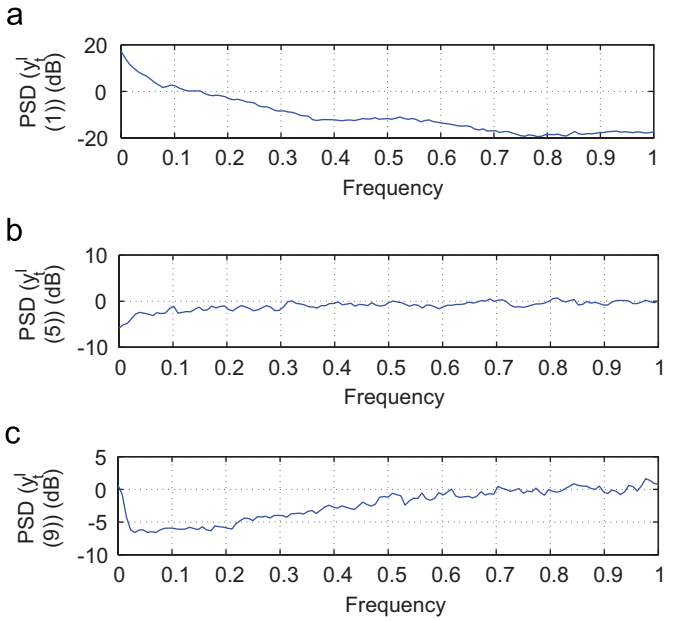

Fig. 3. Power spectral density for independent components number 1 (a), 5 (b) and 9 (c) of the Lena image for $k = 3$.

DFT, DCT or DWT [16] to later compute its inverse transform as $\boldsymbol{x}_t^W$. At detection, we recover the watermark using ICA. In this approach, we need a second mixture of these two sources to perform its separation using ICA as in (1)–(2). We may propose the additional mixture to be the cover image i_t itself. Therefore, the mixing model in (1) results as

$$\begin{bmatrix} \boldsymbol{x}_t^I + \boldsymbol{x}_t^W \\ \boldsymbol{x}_t^I \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_t^W \\ \boldsymbol{x}_t^I \end{bmatrix}. \tag{9}$$

In the next step, we compute matrix $\boldsymbol{B}$ in (2) for this mixture. Notice that this matrix should ideally be the inverse of the mixing matrix in (9). If the projection matrix $\boldsymbol{B}$ has just one non-zero entry per column and per row we have no watermark detected at all. Otherwise, we have the cover image and an estimation of the watermark, up to scaling and permutations, as

$$\begin{bmatrix} \hat{\boldsymbol{x}}_t^W \\ \boldsymbol{x}_t^I \end{bmatrix} = \boldsymbol{B} \begin{bmatrix} \boldsymbol{x}_t^I + \boldsymbol{x}_t^W \\ \boldsymbol{x}_t^I \end{bmatrix}. \tag{10}$$

Notice that this method is basically a non-blind detection algorithm. It is remarkable how it outperforms (see [16]) the simpler method based on subtracting the cover image from the watermarked one [10].

On the contrary, we could think of using the original watermark as second mixture in (10). This would lead us to a blind method. However, since after separation we obtain a mixture of the original and the detected watermarks we do not have a valid estimation of this last one. Some methods have been proposed to partially overcome this problem [28]. In any case, it is interesting to notice that we improve a simple subtraction whenever we have simple manipulations of the image such as scaling. Furthermore, the key of this approach consists in going further than checking for correlation between the watermark and the watermarked image, since it also involves computing higher-order cross-moments to detect the existence of the watermark. This can be studied by checking the entries of the projection matrix $\boldsymbol{B}$. Methods presented above assume that the cover image, or its transform coefficients, and the watermark are statistically independent. Besides, these algorithms may not be regarded as watermarking algorithms as they are just ICA-based watermark detectors. In the following sections we propose the use of ICA as a transform domain where to embed the watermark [20]. We show that this allows a blind detection and estimation of the watermark.

4. A general ICA-based watermarking algorithm

We devote this section to the general design of a ICA-based watermarking algorithm. We develop the embedding and the extraction methods.

4.1. Embedding

Fig. 4 displays the block diagram of a embedding algorithm for ICA. We propose the following steps:

Algorithm 1. *Embedding*:

1. Image to components. *Compute components of the cover image* $\boldsymbol{I}$ *as* $\boldsymbol{x}_t^I = \Upsilon(\boldsymbol{I}, k)$ *following* (6)–(7).

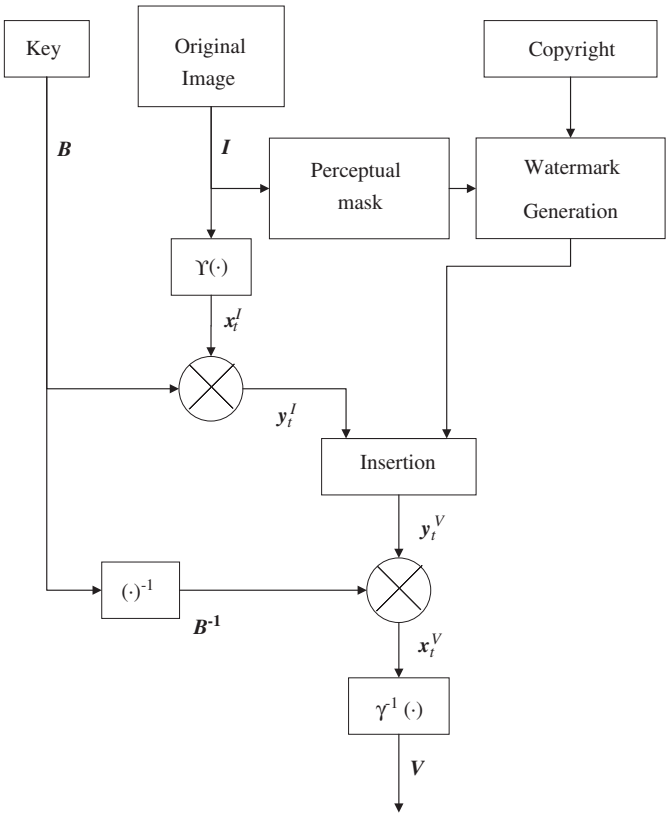

Fig. 4. General embedding algorithm for ICA-based watermarking.

2. ICA components. *Compute the ICA components* $\boldsymbol{y}_t^I = \boldsymbol{B}\boldsymbol{x}_t^I$ *in* (8) *by using an ICA projection* $\boldsymbol{B}$, *the key of the insertion method.*
3. ICA watermarked image components. *Compute components* $\boldsymbol{y}_t^V$ *by embedding the watermark* $\boldsymbol{W}$ *in the ICA components* $\boldsymbol{y}_t^I$. *A perceptual mask may be used to generate the watermark.*
3. Watermarked components to image. *Restore the watermarked image* $\boldsymbol{V} = \Upsilon^{-1}(\boldsymbol{x}_t^V)$ *from components* $\boldsymbol{x}_t^V = \boldsymbol{B}^{-1}\boldsymbol{y}_t^V$.

4.2. Detection

The watermark detection yields:

Algorithm 2. *Detection*:

1. Watermarked image to watermarked components. *Compute the components* $\boldsymbol{x}_t^V = \Upsilon(\boldsymbol{V})$ *of the watermarked image* $\boldsymbol{V}$ *using* (6)–(7).
2. Watermarked components to ICA components. *Compute the independent components* $\boldsymbol{y}_t^V$ *of the image as* $\boldsymbol{y}_t^V = \boldsymbol{B}\boldsymbol{x}_t^V$.
3. Extraction. *Extract the watermark from* $\boldsymbol{y}_t^V$.
4. Detection. *Estimate the probability of watermark detection* (*or false alarm*) *and retrieve the copyright information.*

As described in the following section, a particular design of step 3 of the detection algorithm yields a blind approach. In this blind method, we need no other information than a key, matrix $\boldsymbol{B}$, to extract the watermark.

4.3. Discussion

One of the main drawbacks of the method is its spatial dependence, as the scope of the transformation is limited to each $k \times k$ block in the cover image. This makes it difficult to face spatial transformations such as a resize of the image. However, one of the main advantages of the proposed watermarking algorithm is that it may be public, as its robustness rests on the knowledge of the key $\boldsymbol{B}$. This matrix may be computed by using the results in [6,15], i.e., by using other images belonging to the same class. Therefore, we may use the same ICA projection in the watermarking of a set of images. We may conclude that this new method better conceals the watermark based on the principle by Kerckhoffs [14]. Notice that in some other transform domain watermarking methods, such as DCT, anyone can easily access the transform coefficients.

Some of the blocks of the general embedding and detection methods above may be developed in several ways. In the following section, we focus on the generation, insertion and extraction of the watermark to endow the method with blindness and other remarkable features. In the non-blind case we may use a similar scheme [21].

5. A practical RW algorithm

In this section, we face the generation of the watermark, its embedding in the cover image using a perceptual mask, the watermark extraction and the computation of the probability of detection.

5.1. Watermark generation

In RW we face the detection of a watermark from a severely attacked watermarked image. This process involves the estimation of the probability of detection (or false alarm) and the retrieval of the copyright information to correctly identify the ownership of an image. Besides, we want the watermark to be as imperceptible as possible. These requirements are not fulfilled with any image as watermark, as proposed in [20]. Nevertheless, if we introduce a noise-like watermark, a simple correlation between the original and the extracted watermark may be used as detection algorithm. Furthermore, the magnitude of the correlation peak may be used in the estimation of the probability of detection. Notice that a simple logo or any other image may be embedded as watermark after a scrambling process. In this paper, we propose to use a noise-like image, the spreading code, modulated with a short message [10,11]. The spreading code was generated as a sequence of white Gaussian noise. In this scheme, the watermark can be considered as a weak signal transmitted through a very noisy channel, the cover image plus the attacks. In addition, we pay special attention to methods hiding every bit of the message over the entire image [18] (“holographic” property [5]) to make the method robust to cropping-based attacks [18].

In order to *spread* the watermark over the entire image we use a circular convolution as follows. We propose the watermark $\boldsymbol{W}$ to have the size of one component, i.e., $n/k \times m/k$. The watermark $\boldsymbol{W}$ is computed as

$$\boldsymbol{W} = \boldsymbol{P} \otimes \boldsymbol{Q}, \tag{11}$$

where $\boldsymbol{P}$ is key-dependent pseudo-random image, $\boldsymbol{Q}$ an image containing the bits of the message and $\otimes$ denotes circular convolution. Let $\boldsymbol{M}$ be a $p \times p$ matrix with binary-pixels, the bits of the message. We compute matrix $\boldsymbol{Q}$ as follows:

$$\boldsymbol{Q}(i,j) = \sum_{r,s} M(r,s)\delta(i - r \cdot n_p/2, j - s \cdot m_p/2), \tag{12}$$

where $n_p = n/(k \cdot p)$ and $m_p = m/(k \cdot p)$. Thus, matrix $\boldsymbol{Q}$ is an $n/k \times m/k$ zero-valued matrix except for the bits of the message, located at the center of each $n_p \times m_p$ block. Finally, note that the convolution in (11) may be easily carried out by means of the 2-dimensional DFT.

5.2. Watermark embedding

We now focus on the insertion process in Fig. 4. We first decompose the cover image into its components

$\boldsymbol{x}_t^I = \Upsilon(\boldsymbol{I}, k)$ as defined in (6)–(7). Given the key, matrix $\boldsymbol{B}$, we may compute the independent component as $\boldsymbol{y}_t^I = \boldsymbol{B}\boldsymbol{x}_t^I$, where the components $y_t(i)$, $i = 1, \ldots, k^2$, have been arranged in descending order of magnitude, i.e., variance. At this point of the embedding algorithm we may propose to introduce the watermark into the r last ICA components. However, these usually are the high frequency components. As the watermark is intended to survive common image manipulations, such as JPEG compression, this is not a practical method. As discussed in Section 2, the highest magnitude independent components are the ones associated to the edges of the image. These are those we retain when using ICA to compress or encode images. Therefore, and similarly to other frequency transform watermarking algorithms [10], we propose to insert the watermark into these components.

In the last subsection we proposed the watermark $\boldsymbol{W}$ in (11) to have the size of any of the image components, $n/k \times m/k$. Therefore, $\boldsymbol{y}^W = \Upsilon(\boldsymbol{W}, 1)$ gives us a row vector, the watermark reshaped as one component. The components of the watermarked image yield:

$$y_t^V(h) = y_t^I(h) + \alpha_h y_t^W, \quad h = r_l, \ldots, r_t;$$
$$y_t^V(h) = y_t^I(h), \quad h \neq [r_l, r_t], \tag{13}$$

where α_h is a scaling factor to control the perception of the watermark and we set $r_l = 2$ to exclude the DC component. We can view α_h as a relative measure of how much we must alter component h to modify the perceptual quality of the image. However, one may have little idea of how sensitive the image is to these values. In all our experiments we used a single parameter $\alpha_h = \alpha$, $\forall h$. By applying the inverse transformation $\Upsilon^{-1}(\cdot, k)$, the embedding stage yields

$$\boldsymbol{V} = \boldsymbol{I} + \alpha\tilde{\boldsymbol{W}}. \tag{14}$$

Note also that in the embedding of the watermark we have added the watermark component to the image. Other techniques such as multiplicative or exponential approaches are possible [10].

5.3. Perceptual mask

A critical feature of the watermark embedding algorithm is the ability to provide a transparent watermark that does not noticeably alter the perceived quality of the content and, at the same time, is maximally robust to attacks. In this sense we may significantly improve the performance of the method by using a perceptual masking. This technique exploits the properties of the human visual system by increasing the levels of the watermark in those areas where it is not perceptible. In this paper, the perceptual mask is an image computed by just performing a simple (3×3) sobel edge detection to the cover image. Later, we multiply the watermark by the perceptual mask before adding it to the cover image. Hence, the watermark at every pixel of the image will be enhanced according to the presence of edges nearby. Recalling Eq. (14), the whole embedding process yields $\boldsymbol{V} = \boldsymbol{I} + \boldsymbol{\Lambda} * \tilde{\boldsymbol{W}}$, where $\boldsymbol{\Lambda}$ is the perceptual mask and $*$ denotes entry-wise matrix multiplication (Hadamard or Schur product). The use of other improved perceptual masking [27] are out of the scope of this paper.

5.4. Watermark detection

In the embedding stage we modulated the copyright message with an approach close to the direct sequence spread spectrum DS–SS technique [7,26]. With DS–SS we discriminate the user of interest in a environment where interference and noise are present. At reception, if the channel is just additive white Gaussian noise (AWGN), the optimum receiver is the MF. This detector maximizes the signal-to-noise ratio by just correlating the received signal and the spreading code (here pseudo-noise) used at the transmitter. In this paper we propose to blindly detect the watermark by using this MF (for a non-blind approach see [21]). Therefore, we simply correlate the spreading sequence $\boldsymbol{P}$ with the watermarked image-independent components in (13). We thus model the ICA components of the cover image as white Gaussian noise. Notice that as the DC component cannot be modeled as noise we did not consider inserting the watermark in this component. Besides, the detector is the optimal solution if the attack is just AWGN. The watermark may be detected as described in Fig. 5. We first compute the ICA components of the watermarked image $\boldsymbol{V}$ as

$$\boldsymbol{y}_t^V = \boldsymbol{B}^{-1}\Upsilon(\boldsymbol{V}, k). \tag{15}$$

Then we sum all components $h = r_l, \ldots, r_t$, improving the signal-(watermark) to-noise (image + attacks) ratio,

$$\hat{\boldsymbol{y}}_t^W = \sum_{h=r_l}^{r_t} \boldsymbol{y}_t^V(h) \tag{16}$$

and reshape the resulting vector into the matrix

$$\hat{\boldsymbol{W}} = \Upsilon^{-1}(\hat{\boldsymbol{y}}_t^W, 1). \tag{17}$$

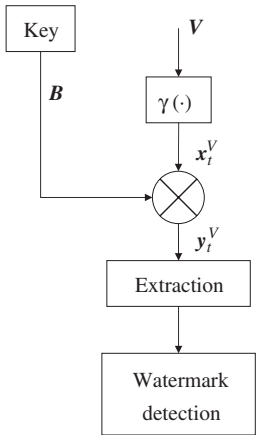

Fig. 5. General detection algorithm for ICA-based watermarking.

Assuming that we have the key to generate the spread sequence $\boldsymbol{P}$ used in the generation of the watermark, we estimate matrix $\boldsymbol{Q}$ in (12) as

$$\hat{\boldsymbol{Q}} = \boldsymbol{P} \otimes \hat{\boldsymbol{W}}. \tag{18}$$

Starting from matrix $\hat{\boldsymbol{Q}}$ we may either estimate the message $\hat{\boldsymbol{M}}$ and its associated bit error rate (BER) or the probability of detection.

It is interesting to remark the main differences between this ICA-based detection algorithm and other transform domains methods. First of all, with ICA we cope with blindness. In DCT-based methods [10] the original image is needed in the detection of the watermark, as the correlation between the transform coefficients of the cover image and the SS watermark (pseudo-noise) does not cancel. On the contrary, the proposed ICA watermarking algorithm embeds the watermark into the independent components of the cover image. Since these components are orthogonal to the watermark we can retrieve the watermark by using a simple MF. On the other hand, if someone knows the transformation used he could base his attack in, e.g., manipulating the transform coefficients where the watermark was embedded. For example, in DCT [10] the main coefficients have different spectral properties than any SS watermark. Hence, a filtering could help removing the watermark. In the ICA approach, even if we have the right key (matrix $\boldsymbol{B}$) of the ICA method, we still cannot easily remove the watermark from the cover image as they have similar, noise-like spectral properties.

5.5. Probability of detection and BER

Similar to other methods [10], we propose to use the circular convolution of the extracted watermark and the original pseudo-random sequence, $\hat{\boldsymbol{Q}}$ in (18), to estimate the copyright message and the probability of detection or false alarm. In the embedding process we used the pseudo-random noise to spread the bits over the entire image. These bits where placed at the central values of every $n_p \times m_p$ block of matrix $\boldsymbol{Q}$. Hence, assuming perfect synchronization, the central values of each $n_p \times m_p$ block of matrix $\hat{\boldsymbol{Q}}$ should be peak values whose signs yield the copyright bits. Besides, we can average the absolute value of the correlation for every bit (block) and then compare the central value to the rest of pixels modeled as white Gaussian noise. This statistical model leads to an easy estimate of the probability of detection. If we face the synchronization problem, we must compute the maximum correlation output for every possible shift. The process yields as follows:

- We first compute the correlation $\hat{\boldsymbol{Q}}^{\alpha,\beta}$ in (15)–(18) for every 2-dimensional shift of the attacked watermarked image $\boldsymbol{V}^{\alpha,\beta}(i,j) = \boldsymbol{V}(i-\alpha, j-\beta)$, where $1 \leqslant \alpha, \beta \leqslant k$.
- We compute the averaged correlation over every $n_p \times m_p$ block as

$$\mathcal{Q}^{\alpha,\beta}(i,j) = \sum_{p=1} \sum_{q=1} |\hat{\boldsymbol{Q}}^{\alpha,\beta}(i+(p-1)n_p, j+(q-1)m_p)|. \tag{19}$$

- We locate the peak values

$$c^{\alpha,\beta}_{\max} = \max_{ij} \mathcal{Q}(i,j)^{\alpha,\beta}. \tag{20}$$

- We select the shift (ζ,η) that provides the maximum averaged correlation peak

$$(\zeta,\eta) : c^{\zeta,\eta}_{\max} \geqslant c^{\alpha,\beta}_{\max} \quad \forall \alpha,\beta. \tag{21}$$

After synchronization, we estimate the received message bits as

$$\hat{\boldsymbol{M}}(p,q) = \hat{\boldsymbol{Q}}^{\zeta,\eta}(i_{\max} + (p-1)n_p, j_{\max} + (q-1)m_p), \tag{22}$$

where $i_{\max}$, $j_{\max}$ are those indexes satisfying $c^{\zeta,\eta}_{\max} = \mathcal{Q}^{\zeta,\eta}(i_{\max}, j_{\max})$.

If we know the original transmitted message we may also easily compute the BER. On the other hand, we assume each entry $z = \{\mathcal{Q}^{\zeta,\eta}(i,j) : (i,j) \neq (i_{\max}, j_{\max})\}$ to be distributed as a Gaussian random variable and we estimate its mean $\overline{z}$ and variance σ_z^2. Then we compute the probability of detection as the probability of every other point different from $c^{\zeta,\eta}_{\max}$ to be lower than $c^{\zeta,\eta}_{\max}$,

$$p_d = F_z(c^{\zeta,\eta}_{\max}, \overline{z}, \sigma_z^2)^{(n_p \cdot m_p - 1)}, \tag{23}$$

where $F_z(c, \overline{z}, \sigma_z^2) = Pb(z \leqslant c)$ is the Gaussian cumulative probability distribution of z. Notice that we do not use the central value $c^{\zeta,\eta}_{\max} = \mathcal{Q}^{\zeta,\eta}(n_p/2, m_p/2)$ but the maximum value for all (i,j) computed after synchronization.

6. Experimental results

We first illustrate the robustness of the method to several attacks. We watermarked ten different gray-scale, in the range [0,1], 512×512 images. We first computed $\boldsymbol{x}_t^I = \mathcal{Y}(\boldsymbol{I}, k=3)$ and then the independent components of the image as $\boldsymbol{y}_t^I = \boldsymbol{B}\boldsymbol{x}_t^I$. We used the separating matrix $\boldsymbol{B}$ computed for the image in Fig. 6 as the key of the method. The watermark was generated as the spread version of a 2-dimensional message of 8×8 bits. We used a second key, a seed, to generate the spreading code as pseudo-random noise. Then, we added the watermark, after a perceptual masking, to the ICA components $h = 2,3,4,5,6$ of the cover image. The final peak signal-to-noise ratio (PSNR) was 41 dB. At this point we must emphasize that for PSNR = 41 dB the visual perception of the watermark in other approaches such as DCT was much more significant than in the ICA method.

Fig. 6. Image used to obtain the ICA projection, matrix $\boldsymbol{B}$.

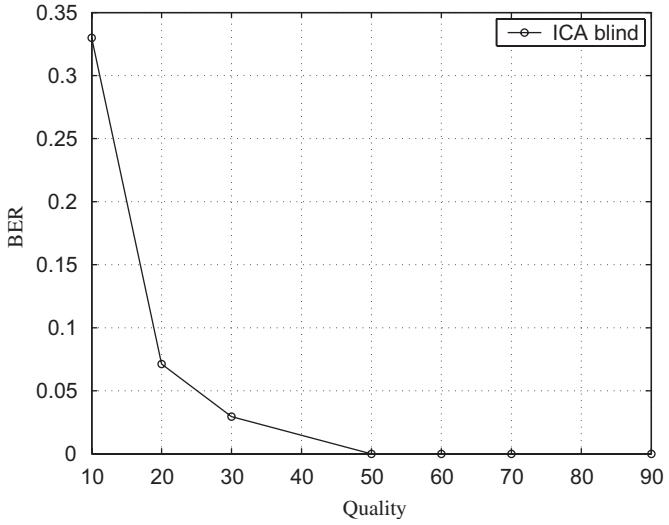

Fig. 7. BER after JPEG-based attack along the compression rate for the Blind ICA algorithm (○).

Table 1
Averaged probability of false alarm (1-probability of detection) and erroneous bits for several attacks on nine gray-scale images

Attack	Prob. of false alarm	Bit error rate (BER). bits
AWGN, $\sigma = 0.15$	0	0.0243
Quantization 2^2	0	0
Median (3×3)	0	0.0104
JPEG 20%	0.9984×10^{-2}	0.0747
Cropping 90%	0.3234×10^{-9}	0.0278
Median (4×4)	0.888	0.5375
Cropping 95%	0.100	0.722

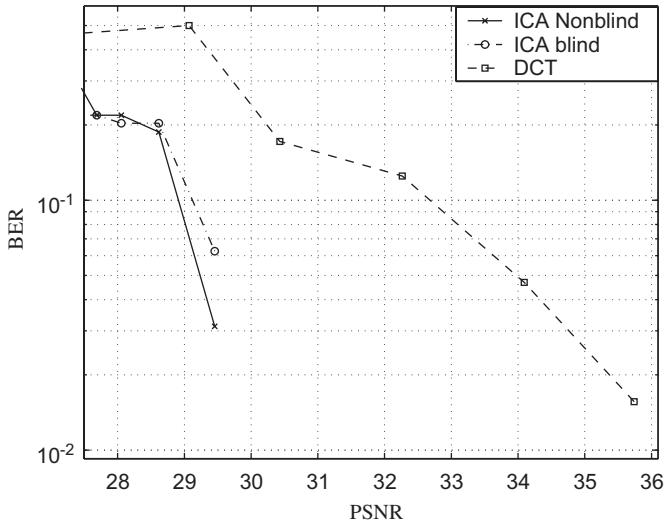

Fig. 8. BER after Wiener filtering-based attack along the PSNR for the (a) non-blind ICA (×), (b) blind ICA (○) and (c) DCT (□) algorithms.

We performed the following different attacks: white additive Gaussian noise with standard deviation $\sigma = 0.15$, quantization of the image to 2^2 levels, 3×3 median filtering, JPEG compression to 20% of the original size and cropping the 90% of the image. The results of the detection algorithm are included in Table 1. The attacks have to highly distort the image to remove the watermark: the noise should have standard deviations over 0.15, cropping has to be about 95%, or the image has to be JPEG compressed to 10% of its original size (see Fig. 7). We conclude that the ICA blind method is robust to noise, quantization and cropping. However, it is sensitive to spatial transformations. As a matter of fact, it does not survive a median filtering of size 4×4.

In Fig. 7 we illustrate the robustness, of the ICA approach in this paper, to JPEG compression.

In Fig. 8 we illustrate the robustness of the methods to attacks based on the knowledge of the watermarking method. We compared the robustness of the method to that of the non-blind DCT approach in [10] when we perform an attack on the transform coefficients. In the DCT approach we generated the watermark in the same way as in (11)–(12), including the perceptual masking described in Section 5.3. The watermark was a 64×64 image, the result of spreading 8×8 bits. We then embedded the watermark into the 64^2 most significant DCT coefficients, excluding the DC. At detection, we subtracted the cover image from the watermarked image, possibly attacked, to later compute the DCT coefficients, extract the watermark and estimate the copyright message.

Since the bits of the copyright information were spread by using white Gaussian noise, we used an adaptive noise-removal algorithm such as the Wiener filtering. In the DCT approach we computed the DCT coefficients and performed a Wiener filtering to the 64^2 coefficients where the watermark was embedded. As we gradually increased the power of the removed noise, the BER and the PSNR of the attacked image deteriorates. In Fig. 8 we may observe that we have a non-zero BER for PSNR under

35 dB. Regarding the ICA methods, we cannot perform the same attack as we do not know the key, the separating matrix $\boldsymbol{B}$, to compute the ICA transform coefficients. Therefore, we must estimate it, e.g., by applying ICA to the watermarked image. In this experiment we propose to apply ICA [22] to the watermarked image to filter the resulting independent components number $h = 2, 3, 4, 5, 6$. We depict the average results for ten images in Fig. 8. Only for low PSNR values we have a poor BER, i.e., we have to completely remove the independent components of the image to destroy the watermark. Finally, it is interesting to remark that the proposed method and the non-blind ICA algorithm in [21] have quite a similar performance, i.e., the correlation between the independent components of the image and the spreading sequence is close to zero.

7. Conclusions

We have proposed a new approach to the robust blind watermarking of digital images based on the ICA image processing in [3,15]. In this algorithm we embed a SS-based watermark into the independent components of the cover image. As these independent components are orthogonal to the spreading sequence, a simple MF is used as blind detector. The robustness of the method rests on two main ideas. On the one hand, the ICA projection used is a key of the method as it is needed to completely remove the watermark. On the other hand, the statistical and spectral characteristics of the independent components are close to that of the spread sequence, i.e., they are noise-like. Therefore, even if we know the ICA projection it is still harder to remove the watermark than in other approaches, where transform coefficients are public and have colored spectra. The experiments included illustrate how this novel blind method successes in extracting the watermark, even when the image has been strongly attacked. The method is robust against additive noise, JPEG compression, quantizing or transform coefficients filtering. Some topics remain as subjects of research. Particularly, approaches to endow the method with robustness against rotation, resize or commercial attacks.

Acknowledgment

Thanks are due to the Spanish government (TIC-2003-03781, MEC TEC2006-13514-C02-02/TCM) and European Union (FEDER) for funding.

References

[1] S.I. Amari, Neural learning in structured parameter spaces—natural Riemannian gradient, in: M.C. Mozer, M.I. Jordan, T. Petsche (Eds.), Advances in Neural Information Processing Systems, vol. 9, The MIT Press, Cambridge, MA, 1997, p. 127.

[2] M. Barni, F. Bartolini, V. Cappellini, A. Piva, A DCT-domain system for robust image watermarking, Signal Process. 66 (1998) 357–372.

[3] A.J. Bell, T.J. Sejnowski, Edges are the independent components of natural scenes, in: M.C. Mozer, M.I. Jordan, T. Petsche (Eds.), Advances in Neural Information Processing Systems, vol. 9, The MIT Press, Cambridge, MA, 1997, p. 831.

[4] S. Bounkong, B. Toch, D. Saad, D. Lowe, Ica for watermarking digital images, J. Mach. Learn. Res. 4 (2003) 1471–1498.

[5] A. Bruckstein, T. Richardson, A holographic transform domain image watermarking method, Circ. Syst. Signal Process. 17 (3) (1998) 361–389.

[6] M.F. Bugallo, A. Dapena, L. Castedo, Image compression via independent component analysis, in: Learning, Leganés, 2000.

[7] A.J. Caamaño, R. Boloix-Tortosa, J. Ramos, J.J. Murillo-Fuentes, Hybrid higher order statistics learning in multiuser detection. IEEE Trans. Man Cybern. C 34 (4) (2004) 417–423.

[8] J.F. Cardoso, Blind signal separation: statistical principles, Proc. IEEE 86 (10) (1998) 2009–2025.

[9] P. Comon, Independent component analysis, a new concept?, Signal Process. 36 (3) (1994) 287–314.

[10] I. Cox, J. Kilian, T. Leighton, T. Shamoon, Secure spread spectrum watermarking for multimedia, IEEE Trans. Image Process. 6 (12) (1997) 1673–1687 URL: ⟨citeseer.nj.nec.com/cox95secure.html⟩.

[11] J.R. Hernández, F. Pérez-González, The impact of channel coding on the performance of spatial watermarking for copyright protection, in: Proceedings of the ICASSP'98, vol. V, Seattle, USA, 1998, pp. 2973–2976.

[12] J.I.A.S. Hornillo-Mellado, R. Martín-Clemente, C.G. Puntonet, Application of independent component analysis to edge detection and watermarking, in: Proceedings of the IWANN'03, Mahón, Spain, 2003, pp. 273–280.

[13] A. Hyvrinen, J. Karhunen, E. Oja, Independent Component Analysis, Wiley, New York, 2001.

[14] A. Kerckhoffs, La cryptographie militaire, J. Sci. Mil, IX: 5–38 (1883) 161–191.

[15] T. Lee, M. Lewicki, T. Sejnowski, Unsupervised classification with non-gaussian mixture models using ICA, in: Advances in Neural Information Processing Systems, vol. 11, The MIT Press, Cambridge, MA, 1999, pp. 58–64.

[16] J. Liu, X. Zhang, J. Sun, M.A. Lagunas, A digital watermarking scheme based on ICA detection, in: Proceedings of ICA2003, Nara, Japan, 2003, pp. 215–220.

[17] S. Makeig, T.-P. Jung, D. Ghahremani, A. Bell, T. Sejnowski, Blind separation of auditory event-related brain responses into independent components, Proc. Nat. Acad. Sci. USA (1997) 10979–10984.

[18] I. Mora-Jimenez, A. Navia-Vazquez, A new spread spectrum watermarking method with self-synchronization capabilities, in: Proceedings of the ICIP2000, Vancouver, BC, Canada, 2000.

[19] J. Murillo-Fuentes, F. González-Serrano, Median equivariant adaptive separation via independence: application to communications, Neurocomputing 49 (1) (2002) 389–409.

[20] J. Murillo-Fuentes, H. Molina-Bulla, F. González-Serrano, Independent component analysis applied to digital image watermarking, in: Proceedings of the ICASSP'01, vol. III, Salt Lake City, USA, 2001, pp. 1997–2000.

[21] J.J. Murillo-Fuentes, Independent component analysis in the watermarking of digital images, Lecture Notes on Computer Science. vol. 3195, 2004, pp. 938–945.

[22] J.J. Murillo-Fuentes, F.J. Gonzlez-Serrano, A sinusoidal contrast function for the blind separation of statistically independent sources, IEEE Trans. Signal Process. 52 (12) (2004) 3459–3463.

[23] N. Nikolaidis, I. Pitas, Copyright protection of images using robust digital signatures, in: Proceedings of ICASSP'96, ⟨http://poseidon.csd.auth.gr/signatures/⟩, Atlanta, GA, 1996, pp. 2168–2171.

[24] S.E. Noel, H.H. Szu, Multimedia authenticity with ICA watermarks, in: H.H. Szu, M. Vetterli, W.J. Campbell, J.R. Buss (Eds.),

Proceedings of the SPIE, Wavelet Applications VII, vol. 4056, 2000, pp. 175–184.
[25] F. Theis, A new concept for separability problems in blind source separation, Neural Comput. 16 (2004) 1827–1850.
[26] S. Verdú, Multiuser Detection, Cambridge University Press, Cambridge, NY, 1998.
[27] R. Wolfgang, C. Podilchuk, E.J. Delp, Perceptual watermarks for digital images and video, Proc. IEEE 87 (7) (1999) 1108–1126.
[28] D. Yu, F. Sattar, A new blind watermarking technique based on independent component analysis., in: F.A.P. Petitcolas, H.J. Kim (Eds.), IWDW, Lecture Notes in Computer Science, vol. 2613, Springer, Berlin, 2002, pp. 51–63.

Juan José Murillo-Fuentes was born in Sevilla in 1973. He received his Telecommunications Engineering degree in 1996 from the Universidad de Sevilla, Spain, and his Ph.D. degree in Telecommunication Engineering in 2001 from the Universidad Carlos III de Madrid, Spain. He is currently an Associate Professor at the Department of Signal Theory and Communication, Universidad de Sevilla. His research interests lie in algorithm development for blind source separation, Gaussian processes and other signal processing tools, and their application to digital communications and image processing.