Digital Image Watermarking using Mixing Systems

G.Voyatzis and I.Pitas
Department of Informatics
University of Thessaloniki, Thessaloniki, 54006 GREECE
fax: +3031-996304, email: voyatzis,pitas@zeus.csd.auth.gr

Abstract

This paper presents a watermarking scheme for copyright protection of digital images. A binary logo is the copyright label which is embedded in grayscale or color digital images. A set of integer parameters, selected by the legal owner, controls the watermarking algorithm via a strongly chaotic (mixing) system. Watermark detection is performed without resorting to the original image. The embedded binary logo is reconstructed or the statistical detection certainty is provided indicating the watermark existence. Numerical experiments testify the efficiency of a particular watermarking algorithm as a reliable verification tool for proving copyright ownership of the digital image.

1 Introduction

Copyright protection of digital images, audio and video, is a novel and very interesting research topic. The technology of digital services grows rapidly and distributed access to such services through computer networks is a matter of urgency. However, network access does not protect the intellectual property rights on digital products which can be reproduced and used illegally. An efficient way to solve this problem, is to use *watermarks* [1]-[5]. One approach to watermarking is to consider a "secret" code, described by a digital signal, carrying information about the copyright holder of the product. The watermark is embedded in the digital data by using a data hiding technique, and it is perceptually unnoticed. The copyright holder is the only person who can demonstrate the existence of its own watermark, and subsequently, proves or strongly indicates the origin of the product. However, watermark will be an effective and valuable tool for protecting intellectual property rights, if its removal from digital data, or its replacement by a forged one, is as difficult as possible and is followed by a significant reduction of the perceived quality of the digital product. Therefore, watermarks should remain invariant to a large class of modifications such as filtering, compression, attacks etc.

Watermarks can be applied to digital data of still images, audio and video. This paper refers exclusively to digital grayscale or color image watermarking. For this important particular case, we state that the following requirements should be satisfied :

1. Watermark should be perceptually invisible. Alterations introduced in the image should not reduce its perceived quality.
2. Alterations, introduced in images by the watermark, should not be determined and, subsequently, easily removed by a third person. The possession of a key must be the necessary condition to proceed to watermark detection.
3. A sufficient number of watermarks in the same image, all detectable by their own key, can be produced. The same key can be used by a producer to watermark a large set of images.
4. Watermarks should be statistically invisible. The possession of a large set of images, watermarked by the same watermark, should not help an attacker to determine

the watermark.
5. The detection of the watermark should not require access to the original image data. This demand is necessary for avoiding time consuming search in large digital image libraries and for applying "web-crawling" detection.
6. Watermark should be robust as much as possible against attacks or image processing operations that preserve a desired image quality.

Until now, the proposed algorithms do not satisfy completely the above requirements. The watermarks either are very sensitive to image modifications [1, 2], or they do not satisfy requirements (3) or/and (4) [6, 7] or they need the original image for detection [8]. Every watermark should resist, as much as possible, to image modifications including format conversion, compression, filtering, scaling etc.

The watermark detection is based on a parameter (or a set of parameters) that represents the private *key* of the person who watermarks the image (usually the owner himself or an authorized person in the distribution channel). The possession of the *key* is the necessary condition to detect the watermark. Watermark casting by using different keys should provide uncorrelated alterations on the image. The detection of a particular watermark should be performed if and only if the suitable key is used. The set of keys that are suitable for watermarking, should be quite large in order to serve all the creators of digital products and discourage pirates to define the key by trial and error procedures. Preferably, embedding and detection algorithms should be publicly known or they should applied by trusted third parties [3].

Multiple watermarking on the same image must be accounted for, since it may be desirable feature for tracing images in distribution channels [2]. Generally, multiple watermarking can not be prevented. The legal owner is the one who possesses or can produce a copy of the image not containing the other watermarks. This is true when counterfeit watermarks, pointed out by Craver *et al* [9], can not be produced. Legal proof may be combined by a deposition of the original (non-watermarked) image in a copyright authority before distribution [3].

Alterations in digital data, introduced by the watermarking algorithm, should be small in order to be perceptually invisible. Excessive image alterations either can be detected or may degrade the perceived image quality. However, the greater the alterations the greater the watermark robustness against processing and piratical attacks. Alterations can take place directly in the spatial domain of the image [1, 2, 5, 10]. In this case a set of pixels or blocks is selected and their intensity levels are slightly modified. Correlation or statistical tests are used to detect the watermark. The DCT coefficients [6, 11, 7] of some image blocks (usually of size 8×8) can be used as the watermark embedding space. Cox et al [8] proposed spread spectrum techniques to embed watermarks in still images, audio and video.

The proposed watermarking scheme, is applied to the intensity or luminance domain of a digital image. We hide secretly a publicly known copyright *logo* (i.e. a small BW image representing the legal owner) in the image. The logo can be built up again, by using the watermarked image, revealing the original copyright information. The encryption and the reconstruction of the binary watermark is succeeded by a transformation based on chaotic systems called *toral automorphisms*. The mixing property of such systems guarantees the secure encryption of the copyright information carried by the watermark. In the next Section a brief description of the main properties of toral automorphisms is presented and a suitable form is derived which is directly applied on digital images. In Section 3, the fundamental steps of the embedding and detection algorithms are presented. The structure of the watermarking key, which is image dependent, is described. In Section 4, hypothesis testing is proposed to provide the statistical certainty for the watermark detection. Finally, in Section 5, simulation experiments of a particular algorithm are presented indicating its performance.

2 Toral Automorphisms and Integer lattices

A two dimensional "toral automorphism" can be considered as spatial transformation of planar regions [13]. It is represented by a map:

$$\mathcal{A} : U \rightarrow U \ , \ U = [0,1) \times [0,1) \subset R^2$$

and is defined by the formula:

$$\mathbf{r}' = \mathbf{A}\,\mathbf{r}(mod\,1) \ , \ \begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} a_{11} \ a_{12} \\ a_{21} \ a_{22} \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} \ (mod\,1) \tag{1}$$

where $a_{ij} \in Z$, $det\mathbf{A} = 1$ and $\lambda_{1,2} \notin \{-1,0,1\}$ are the eigenvalues of $\mathbf{A}$. Iterated actions of $\mathcal{A}$ on a point $\mathbf{r}_0 \in U$ form a dynamical system $\mathcal{A}^{(n)} : U \rightarrow U$, given by the iterative process:

$$\mathbf{r}_{n+1} = \mathbf{A}^n \mathbf{r}_0 \ \ (mod\,1) \quad \text{or} \quad \mathbf{r}_{n+1} = \mathbf{A}\mathbf{r}_n \ \ (mod\,1) \tag{2}$$

where $n = 0, 1, 2, ...$ The set of points $\mathcal{O}(\mathbf{r}_0) = \{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2, ...\}$ is an orbit of the system. Automorphisms belong in a special class of Anosov diffeomorphisms which are strongly chaotic systems obeying local instability, ergodicity with mixing, and decay of correlation [13]. Roughly speaking, if V_0 is a dense subset of U then its image V_n under the map $\mathcal{A}^{(n)}$ spreads chaotically over the entire space of U while preserving its area, because $det\,\mathbf{A} = 1$. The *Cat map* $(a_{11} = a_{12} = a_{21} = 1, a_{22} = 2)$ is a classical mixing system in dynamics and an example of its performance is shown in Figure 1.

Although system (2) is strongly chaotic it possesses a dense set of periodic orbits. An orbit $\mathcal{O}(\mathbf{r}_0) = \{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2, ...\}$ is periodic, if it is finite, i.e., there exists a number T of iterations such that $\mathbf{r}_0 = \mathbf{r}_T$. The necessary and sufficient condition for an orbit to be periodic is that the initial position $\mathbf{r}_0$ to have rational coordinates [13] :

$$\mathbf{r}_0 \ = \ \left(\frac{p_1}{q_1} \ , \ \frac{p_2}{q_2} \right) \ \in U$$

where p_i, q_i are coprime integers. We consider the discrete subset of U :

$$\bar{U} = \{(x,y) | x = k/N, y = l/N, k, l \in \{0, 1, .., N-1\} \}$$

where N is the least common multiple of q_1, q_2. If $\mathbf{r}_0 \in \bar{U}$, then $\mathcal{O}(\mathbf{r}_0)$ is periodic, all its elements belong to $\bar{U}$. We consider the following map:

$$\mathcal{A}_N \ : \ L_N \rightarrow L_N \ , \ \mathbf{r}' \ = \ \mathbf{A}\,\mathbf{r} \ (mod\,N) \tag{3}$$

where:

$$L_N \ = \ \{(k,l) \ | \ 0 \leq k < N, 0 \leq l < N\}$$

is an integer lattice (a mesh) of size N. In a similar manner we can have iterated actions of map $\mathcal{A}_N$ according to (2), thus forming a dynamical evolution. We can easily derive the following equivalence relation between the orbits of $\mathcal{A}$ in $\bar{U}$ and the orbits of $\mathcal{A}_N$ in L_N:

If $\mathcal{O}(\mathbf{r}_0) = \{\mathbf{r}_0, \mathbf{r}_1, ..., \mathbf{r}_T\} \subset \bar{U}$ *is a periodic orbit of* $\mathcal{A}$*, then the orbit* $\mathcal{O}(N\,\mathbf{r}_0) = \{N\,\mathbf{r}_0, N\,\mathbf{r}_1, ..., N\,\mathbf{r}_T\}$ *is a periodic orbit of* $\mathcal{A}_N$ *and vice versa.*

All the orbits of the map $\mathcal{A}_N$ are unstable periodic orbits since the eigenvalues $\lambda_1 = 1/\lambda_2$ of the matrix $\mathbf{A}$ are positive. Their periods depend on the parameters a_{ij} of $\mathbf{A}$ and the size N of the lattice L_N. They follow the same rules as the periods of the periodic orbits of automorphisms in a real space which are studied in detail in [14]. Based on the properties of periodic orbits, we can state the following corollary [15]:

For any integer lattice L_N of size N, there is an integer $P = P(a_{ij}, N)$ such that:

$$\mathcal{A}_N^P \mathbf{r} = \mathbf{r} \ (mod\, N) \ , \ \ \forall \mathbf{r} \in L_N \tag{4}$$

We call the integer P *recurrence time*. The elements $\mathbf{r}_i$ of a periodic orbit are distributed quite randomly in L_N. If N is not a prime, ideal (symmetric in some sense) sublattices exist and some classes of orbits lay on them. However, this feature is lost by the composite mixing procedure used in this paper and described in Section 3.1.

The evolution of the orbits depends exclusively on the eigenvalue λ_1 (or λ_2) [14]. Subsequently, automorphisms are one-parameter systems. The parameters a_{ij} in (1) are not independent but are restricted by the relations $det\, \mathbf{A} = 1$ and $trace\, \mathbf{A} = a_{11} + a_{22} = f(\lambda_1)$. We propose a pure one-parameter family of maps $\mathcal{A}_N(k)$ given by the formula :

$$A_N(k) : L_N \rightarrow L_N \ , \ \ \begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} \ (mod\, N) \tag{5}$$

where $(x_n, y_n) \in L_N$ and $k \in [1, N) \subset Z$. The greatest eigenvalue of the matrix is $\lambda_1 = 1 + 0.5(k + (k^2 + 4k)^{1/2})$ and is real and positive for any $k > 0$.

3 The watermarking algorithm

Let I be an original grayscale image of size $N \times N$ represented as :

$$I = \{x_{ij} \mid 0 \leq i < N, 0 \leq j < N, \ x_{ij} \in \{0, 1, ..., G\}\} \tag{6}$$

where G is the number of intensity levels (e.g. $G = 256$). We consider a BW image of size $M_1 \times M_2$

$$W = \{w_{kl} \mid 0 \leq k < M_1, \ 0 \leq l < M_2 \ , \ w_{kl} \in \{0, 1\}\} \tag{7}$$

which is the original form of the watermark representing the copyright label, e.g., a company logo. W should be sufficiently small, i.e.

$$max(M_1, M_2) < N \ \ , \ \ M_1 \cdot M_2 \ll N^2 \tag{8}$$

For color images, x_{ij} denotes the luminance of the pixel (i, j). We assume square images of size $N \times N$ just for simplicity in our presentetion. The method can be extended to include images of any size.

The embedding of W in I takes place by altering the intensity x_{ij} of the pixels according to w_{kl}. The secure encryption of the watermark is based on a chaotic map between the watermark pixels (k, l) and the image pixels (i, j). The basic steps of the algorithm, as described in subsequent subsections, are i) suitable choice of the embedding positions in the image ii) watermark embedding in a digital image domain and iii) watermark detection by reconstructing the copyright logo. The watermark key is formed by some integer parameters. As shown in 3.4, the same key produces different watermarks for different images.

3.1 Pixels selection for mapping the watermark

The map (5) is applied directly to a square image I or to a square subblock $U^{(0)} \subseteq I$ of size $N \times N$. An iterative procedure produces a sequence of mixed blocks :

$$U^{(i)} = \mathcal{A}_N^i(k)\, U^{(0)} \ , \ i = 1, 2, ..., P - 1 \tag{9}$$

where P is the recurrence time. The reconstruction of $U^{(0)}$ from $U^{(n)}$ is performed by applying the inverse automorphism, which always exists :

$$U^{(0)} = \mathcal{A}_N^{-n}(k)\, U^{(n)} = \mathcal{A}_N^{P-n}(k)\, U^{(n)} \tag{10}$$

Various procedures can be designed to mix the watermark (7) by getting a cryptographically secure map between the pixels of the watermark and the locations of the pixels in the image. Subsequently, we present a complete procedure, which encrypts the original watermark and shows the positions for embedding.

First, we consider the watermark W to be a subset of a larger area:

$$W \subset \tilde{W} = \{\tilde{w}_{ij}, 0 \leq i, j < M\}$$

with $max(M_1, M_2) \leq M < N$, where N is the side size of image (6) and $M_1 \times M_2$ is the size of watermark (7). This step is described by the map :

$$\mathcal{T}_\mathbf{0} : W \to \tilde{W} \ , \ \tilde{w}_{i,j} = \begin{cases} w_{ij} & \text{if } i < M_1 \text{ and } j < M_2 \\ Null & \text{otherwise} \end{cases} \tag{11}$$

"*Null*" represents any value (except 0 or 1), for the pixels which do not belong to the original watermark. W spreads in $\tilde{W}$ by applying the map $\mathcal{A}_M^{n_1}(k_1)$ (first mixing) :

$$W' \ = \ \mathcal{A}_M^{n_1}(k_1)\, \tilde{W} \tag{12}$$

W' is placed in a square domain W'' of size $N \times N$, and, at position determined by the vector $\mathbf{s} = (s_1, s_2)$. This is described by the map $\mathcal{T}_s$:

$$\mathcal{T}_s : W' \to W'' \ , \ w''_{i,j} = \begin{cases} w_{i-s_1, j-s_2} & \text{if } s_1 \leq i < M \ , \ s_2 \leq j < M \\ Null & \text{otherwise} \end{cases} \tag{13}$$

A second mixing is applied on W'' and we get the final mixed watermark $\hat{W}$ which spreads in entire $N \times N$ domain:

$$\hat{W} \ = \ \mathcal{A}_N^{n_2}(k_2)\, W'' \tag{14}$$

The values 0, 1,*Null* of the elements $\hat{w}_{ij}$ split $\hat{W}$ in three subsets:

$$\hat{W} \ = \ \hat{W}_0 \cup \hat{W}_1 \cup \hat{W}_{Null}$$

The pixels of $\hat{W}$ are mapped by identity to the image I, thus indicating the pixels of the image where the watermark will be embedded.

The global mapping between the watermark and the image is represented by the transformations (11-14) and is written as:

$$\mathcal{F} : W \to I \ \ , \ \ \mathcal{F} = \mathcal{A}_N^{n_2}(k_2) \circ \mathcal{T}_\mathbf{s} \circ \mathcal{A}_M^{n_1}(k_1) \circ \mathcal{T}_\mathbf{0} \tag{15}$$

The inverse procedure is composed by using the inverse maps in inverse order :

$$\mathcal{F}^{-1} : I \to W \ \ , \ \ \mathcal{F}^{-1} = \mathcal{T}_\mathbf{0}^{-1} \circ \mathcal{A}_M^{P_1 - n_1}(k_1) \circ \mathcal{T}_\mathbf{s}^{-1} \circ \mathcal{A}_N^{P_2 - n_2}(k_2) \tag{16}$$

where $\mathcal{T}_\mathbf{s}^{-1}$ truncates the pixels $(s_1 + i, s_2 + j) \in W''$, $(i, j) \in [0, M)^2$, to form W'. Also the map $\mathcal{T}_\mathbf{0}^{-1}$ truncates the pixels $(i, j) \in [0, M_1) \times [0, M_2) \subset \tilde{W}$ which belong to the original watermark. We note that the components of the global maps $\mathcal{F}$ and $\mathcal{F}^{-1}$ do not commute.

As an example, the logo "rabbit" is used to be the binary watermark of size 51×51 shown in Figure 2a. Figures 2b and 2c represent a typical first and second mixing respectively. The inverse procedure reconstructs the "rabbit" from the chaotically distributed points of Figure 2c.

3.2 Watermark embedding

Let $\mathbf{r} = (r_1, r_2)$ denote the location of a pixel in the watermark W and $\mathbf{p} = (p_1, p_2) = \mathcal{F}(\mathbf{r})$ the corresponding pixel location in the original image I. We alter the intensity $x_\mathbf{p}$ of the pixel $\mathbf{p}$ in I and we get the watermarked image :

$$I' = \{x'_{ij} \mid 0 \leq i, j < N, x_{ij} \in \{0, 1, ..., G\}\} \tag{17}$$

The pixels $\mathbf{p}$ of the image preserve their intensity when $\hat{w}_\mathbf{p} \in W_{Null}$ or, equivalently, when they are not mapped to the original watermark (7):

$$x'_\mathbf{p} = x_\mathbf{p} \quad \text{if} \quad \mathbf{r} = \mathcal{F}^{-1}(\mathbf{p}) \notin W \tag{18}$$

For pixels mapped to watermark, alterations take place according to the bi-valued pixel $\mathbf{r}$ in W. Therefore, the watermark embedding should be performed by applying two different (publicly known) embedding functions g_0 and g_1 which provide the new intensity value for the watermarked pixel :

$$x'_\mathbf{p} = \begin{cases} g_0(x_\mathbf{p}, B_\mathbf{p}, q_0) & \text{if } w_\mathbf{r} = 0 \\ g_1(x_\mathbf{p}, B_\mathbf{p}, q_0) & \text{if } w_\mathbf{r} = 1 \end{cases} \tag{19}$$

$B_\mathbf{p}$ is an image block, i.e. it represents the intensities x_{ij} of a set of pixels that belong to the neighbourhood of the pixel $\mathbf{p}$. The parameter q_0 is positive and controls the degree of difference between the original and the watermarked pixels. It is associated to a "detection" function $D_\mathbf{p}$ defined as follows:

$$D_\mathbf{p}(x'_\mathbf{p}, B'_\mathbf{p}) = \begin{cases} -q_0 & \text{if } w_\mathbf{r} = 0 \\ q_0 & \text{if } w_\mathbf{r} = 1 \end{cases} \tag{20}$$

The block $B'_\mathbf{p}$ refers to the altered intensities x'_{ij}. Thus, the function $D_\mathbf{p}$ is calculated by considering exclusively the watermarked image (17).

A necessary condition, which should be satisfied by the embedding, is the invisibility of the introduced alterations, i.e.

$$|x'_\mathbf{p} - x_\mathbf{p}| \leq \varepsilon(\mathbf{p}) \quad , \quad \forall \mathbf{p} \in I'$$

where $\varepsilon(\mathbf{p})$ denotes the minimum absolute alteration which is perceptually unnoticed. We can consider as $\varepsilon(\mathbf{p})$ a small constant value (e.g. less than 5 for an 8-bit grayscale image [1, 2]). However, better performance can be achieved when $\varepsilon(\mathbf{p})$ is calculated by using masking models based on the human visual system [12, 16].

The contribution of the intensity value of all the pixels in $B_\mathbf{p}$, for watermarking the pixel $\mathbf{p}$, is necessary in order to produce resistant alterations [17, 18]. As it becomes clear in Sections 3.3 and 4, the detection function should be positive or negative when the embedding function g_1 or g_0 are used respectively. Therefore, watermark robustness is succeeded when $D_\mathbf{p}$ preserves its sign when image modifications take place.

3.3 Detection of the Watermark logo

The reconstruction of the logo watermark W is based on the detection function $D_\mathbf{p}$ calculated on the watermarked image. By considering the sign of function $D_\mathbf{p}$, at every pixel in the $N \times N$ embedding domain of the image, we form the following binary set :

$$Z = \{z_\mathbf{p} \mid \mathbf{p} = (i, j) \in [0, N)^2 \, , \, z_\mathbf{p} \in \{0, 1\}\} \tag{21}$$

where :

$$z_\mathbf{p} = \begin{cases} 1 & \text{if } D_\mathbf{p}(x'_\mathbf{p}, B'_\mathbf{p}) > 0 \\ 0 & \text{if } D_\mathbf{p}(x'_\mathbf{p}, B'_\mathbf{p}) < 0 \end{cases}$$

According to the embedding procedure, the mixed watermark set $\hat{W}$ is related to $D_{\mathbf{p}}$ as follows :

$$\begin{array}{lcl} D_{\mathbf{p}}(x'_{\mathbf{p}}, B'_{\mathbf{p}}) > 0 & \Leftrightarrow & \hat{w}_{\mathbf{p}} = 1 \\ D_{\mathbf{p}}(x'_{\mathbf{p}}, B'_{\mathbf{p}}) < 0 & \Leftrightarrow & \hat{w}_{\mathbf{p}} = 0 \end{array} \tag{22}$$

where $\mathbf{p}$ denotes the altered pixels, i.e. the pixels belong to either $\hat{W}_1$ or $\hat{W}_0$. The residue pixels in the set Z correspond to the subset $\hat{W}_{Null}$. Consequently, Z is written

$$Z \; = \; \hat{W}_0 \cup \hat{W}_1 \cup U$$

where U is an arbitrary $\{0,1\}$-valued set. By applying the inverse mixing procedure $\mathcal{F}^{-1}$ to Z, we take :

$$\mathcal{F}^{-1}(Z) = \; \mathcal{F}^{-1}(\hat{W}_0) \cup \mathcal{F}^{-1}(\hat{W}_1) \cup \mathcal{F}^{-1}(U) \tag{23}$$

The pixels of $\hat{W}_1$ and $\hat{W}_0$ are reorganized by applying $\mathcal{F}^{-1}$ and produce the white (1-valued) and the black (0-valued) regions of the watermark W respectively. The pixels of U, after the application of the mixing component $\mathcal{A}_M^{P_1-n_1}$ of $\mathcal{F}^{-1}$, occupy the domain outside of the watermark region $(0, M_1) \times (0, M_2)$ which is not truncated by $\mathcal{T}_{\mathbf{0}}^{-1}$. Thus, from equation (23) we get :

$$\mathcal{F}^{-1}(Z) = \; \mathcal{F}^{-1}(\hat{W}_0 \cup \hat{W}_1) = W \tag{24}$$

Figure 3 illustrates the reconstruction of the watermark "rabbit" from the set Z of the watermarked image "Lena" shown in Figure 4c. The set Z is mixed by the automorphism $\mathcal{A}_N^{P-n_2}(k_2)$ and results in the set shown in Figure 3b. By truncating the subset of size $M \times M$ pointed by the vector (s_1, s_2) and by applying $\mathcal{A}_M^{P-n_1}(k_1)$ the watermark is reconstructed at the upper left corner (Figure 3c).

3.4 The watermarking key

The watermark is extracted by using only the signed image I' and the inverse mixing map $\mathcal{F}^{-1}$ controlled by the integer parameters M, k_1, n_1, s_1, s_2, k_2, n_2. The *key* for the detection consists essentially of these parameters which should satisfy the following constrain :

$$\begin{array}{c} max(M_1, M_2) \; \leq \; M \; < N \\ 0 < k_1 < M \\ m < n_1 < P(M, k_1) - m \\ s_i + M \leq N, i = 1, 2 \\ 0 < k_2 < N \\ m < n_2 < P(N, k_2) - m \end{array} \tag{25}$$

The parameter m is a minimum number of iterations so that the chaotic mixing is satisfactory (e.g $m \approx 10$). Even the smallest change in the key render it unusable to reconstruct the watermark or a part of it. The embedding of a 32×32 watermark in a 256×256 image, can be applied by using parameters that form more than 10^{10} combinations. The map $\mathcal{F}$ can be designed to include more automorphisms, thus enriching the key by an enormous number of possible combinations.

In the present algorithm the positions of the watermarked pixels are the same for all images having the same size. The binary set Z can be extracted from the watermarked image only by using the detection function D. By possessing a sufficient number n of watermarked images of the same size and watermarked by the same *key*, the corresponding sets $Z^{(i)}$ disclose the embedding positions, because :

$$\bigcap_n Z^{(i)} \; = \; \hat{W}_0 \cup \hat{W}_1 \;\; , \;\; i = 1, 2, ..., n, \; n \geq n_0 \tag{26}$$

The probability for such an attack becomes negligible if the copyright holder is facilitated to use different values of the parameter n_2 from an available domain (n_1^*, n_2^*). When the parameter n_2 is selected randomly by the image owner, the detection algorithm should search the watermark (e.g. using the statistical detection described in Section 4) for every iteration $n_2 \in (n_1^*, n_2^*)$.

It is quite convenient if n_2 is an "image dependent" key parameter [10, 12], i.e. the watermark is formed by taken into account characteristics of the image. Such characteristics should be very robust under modifications so that to provide always the same value for the parameter n_2. Next we propose a procedure which determines an image dependent parameter n_2.

We replace the parameter n_2 in the key by an integer K. By using a random number generator with seed K, we choose n large blocks B_i of the original image I. Let, $\bar{x}_i$, $i = 1, .., n$, are the mean intensities of the blocks B_i respectively. We form the following inequalities :

$$\bar{x}_i \quad > \bar{x}_{i+1} \quad , \quad i = 1...n-1 \tag{27}$$

By assigning, for each inequality, the value 0 or 1 when they are false or true respectively, we produce an $(n-1)$-bit integer Q. The iteration number n_2 for the second mixing, is calculated from Q and according to the constrain (25) is given by:

$$n_2 \;=\; m \;+\; Q \; mod \; (P - m) \tag{28}$$

where $P = P(N, k_2)$ is the recurrence time of the second mixing. Since B_i are large blocks, their mean intensity is not affected significantly by the watermark embedding, filtering or compression. Thus, the inequalities (27) are robust to the above mentioned image modifications and, subsequently, the same value for the parameter n_2 is found when calculations are performed either on the original image or on the particular watermarked or modified image.

4 Statistical detection and algorithm performance

The detection algorithm, described in Section 3.3, attempts to reconstruct the original copyright logo without using any information about the original binary watermark. However, we can make the non restrictive assumption that the mixed watermark $\hat{W}$, which is mapped to the image (or the W and the *key*), is known to the owner since its production requires only the key and the original logo. The watermarked pixels are classified in the two sets $\hat{W}_0$ and $\hat{W}_1$ and the associated alterations in the image are designed to satisfy the relation (22). Therefore, statistical detection can be performed by examining the difference :

$$\bar{\delta} = \bar{\delta}_1 - \bar{\delta}_0 = \frac{1}{P_1} \sum_{\mathbf{p} \in \hat{W}_1} D_{\mathbf{p}} \; - \; \frac{1}{P_0} \sum_{\mathbf{p} \in \hat{W}_0} D_{\mathbf{p}} \tag{29}$$

where P_0, P_1 are the numbers of the pixels in the sets $\hat{W}_0$ and $\hat{W}_1$ respectively. $\bar{\delta}$ is expected to be positive and close to the value $2q_0$ when the watermark exists in the image and approximately zero otherwise. However, when the watermarked image is modified, e.g. by lowpass filtering or compression, generally we obtain $\bar{\delta} < 2q_0$. Hypothesis testing for the difference of means should be applied [19]:

H_0 : There is no watermark ($\bar{\delta} \;=\; 0$)

H_1 : There is watermark ($\bar{\delta} \;\neq\; 0$)

The fact that $\bar{\delta}$ is not zero because of the existence of a watermark and not by chance (i.e. H_1 is accepted) is verified with a certainty level and is examined by using the Student's t-test. The possible errors are

Type I Error. We decide H_1 although H_0 is true, i.e. we detect watermark although there is none (false positive) This error is expressed by the *false alarm probability* (P_{fa}) and coinsides with the *significance* of the test denoted by a ($0 \leq a \leq 1$).

Type II Error. H_0 is decided although H_1 is true, i.e. we do not detect the watermark although it exists (false negative). The probability to make such an error is called *rejection probability* (P_{rej}).

A threshold value a_{thres} for the significance a should be determined in order to decide whether the image is watermarked or not. Watermark detection is considered successful for $a < a_{thres}$. When a_{thres} increases, the probability to reject a watermark, although it exists in an image (type II error), decreases. However, the certainty $1-a$ of a positive watermark detection decreases too. Thus, large values of a_{thres} do not provide a strict decision about the watermark existence. When detection is performed on non watermarked images, P_{fa} is given by the cumulative distribution function of the significance a :

$$P_{fa} \;=\; F_a(a_{thres}) \;=\; Prob\{a \leq a_{thres}\}$$

In practice, the following condition should hold approximately :

$$F_a(a_{thres}) \;=\; a_{thres} \tag{30}$$

A convenient way to measure the performance of the algorithm is to calculate the *detection ratio* defined as

$$detection\ ratio = N_d(a_{thres}) \;/\; N_t \tag{31}$$

where $N_d(a_{thres})$ is the number of cases where the watermark is detected correctly, i.e. when we find $a < a_{thres}$ and the watermark indeed exists. N_t is the number of total experiments.

5 Simulation Experiments

Figure 4a shows the original 8-bit grayscale image “Lena” of size 512×512 pixels. Instead of one ”rabbit” of size 51×51, the image 4b, which consists of nine logos, is mixed and embedded in “Lena” producing the watermarked image 4c. We can state that the watermark is perceptually invisible and the two images are visually indistinguishable. The absolute difference of the original and the watermarked image is shown in Figure 4d. The watermark logo is fully detected when the watermarked image has not been processed. Generally, noisy watermark logos are detected when the watermarked image has been compressed or filtered. Figure 5 shows typical examples of such noisy watermark detection. The final reconstructed logo is formed by extracting the most frequent pixel value observed in the nine detected logos. By using this technique, we increase the probability to determine the binary value of the watermark pixels correctly. It can be noticed that only for high JPEG compression ratios (e.g. above 10:1), the detected watermark becomes barely intelligible. The watermark is also well resistant under moving average and median filter.

When reconstructed watermarks can be recognized visually, the statistical detection, described in Section 4, suggest with great certainty the existence of the watermark. For the noisy watermark logos in Figure 5, the significance of the statistical hypothesis testing was less than 10^{-30}.

We applied the statistical detection to the watermarked “Lena”, by considering 1000 different keys, provided by a pseudo-random number generator (the correct key was not included). The function $F_a(a_{thres})$ of the calculated significance $a_i, i = 1, ..., 1000$, is shown in Figure 6. We can conclude that relation (30) is verified with good accuracy (note that logarithmic scales are used).

Therefore, we can claim that the significance of statistical test indicates the probability of false alarm error with good approximation. For example, if we select $a_{thres} = 10^{-4}$, the probability to detect a watermark although it does not exist (or the wrong key is used) is about 1 per 10^4 trials.

When the image is affected quite strongly by image processing or compression, the binary noise which corrupts the reconstructed watermark may make the logo visually unintelligible. Even in such cases, the statistical detection can provide correct decisions with high certainty. Figure 7 presents the significance a, observed from the hypothesis testing applied to JPEG compressed versions of "Lena". For "Lena" of size 256×256, one sample of the watermark "rabbit" has been embedded instead of nine watermark samples embedded in the image of size 512×512. The plot presents the mean value of $log(a)$ observed from 200 watermarked images using keys provided by a pseudo-random number generator. The value of a indicates the existence of the watermark with great certainty even for high compression ratio when the quality of the compressed image becomes quite low. The detection ratio calculated from 200 watermarked images is shown in Figure 8. The detection is absolutely successful with certainty 99.99%, up to compression ratio 13:1 and 33:1 for the 256×256 and the 512×512 image respectively. We may claim that the performance of the watermarking algorithm increases with the size of the image. Large images provide a lot of space to embed many watermark logos. Therefore, the probability for correct watermark reconstruction (by selecting the most frequent pixel value of the embedded logos) increases and the statistical detection also becomes better since the number of data, taken into account in statistical test, grows. The above conjecture is supported by experiments presented in [18].

Finally we note that multiple watermarking (overwatermarking) is possible. Many coexisted watermarks in the image overlap and their detection certainty is reduced. Figure 9 represents the significance a for the detection of a particular watermark W_0, versus the number of other watermarks embedded in the same image afterwards (the same logo "rabbit" for all watermarks but different embedding keys have been used). We can notice that a decreases as the number of superimposed watermarks increases. However, the quality of the produced watermarked image is reduced also by multiple watermarking, as it is shown by the SNR curves in Figure 9. The watermark embedding creates a noisy pattern which becomes perceptually visible after few watermarking applications (when SNR is about less than 30dB).

6 Conclusions

The watermarking algorithm, described in this paper, follows the general concepts mentioned in the introduction. A binary watermark image represents a logo, which indicates the copyright owner. The mixing procedure, used to map the watermark into the image, is the base for developing a secure watermark embedding and a reliable detection. Alterations are performed on the intensity levels of the pixels or on the luminance values for color images.

The proposed watermarking scheme produces watermark signals that show high resistance to filtering and compression. When the quality of the image is not reduced too much, most of the watermark pixels are reconstructed giving a clear picture of the original logo. For large image distortions the statistical detection can be used instead. The image dependent key parameter prevents the production of a counterfeit watermark and the disclosure of a legal one by examining a large set of images watermarked by the same key. Noise or multiple watermarking do not hamper watermark detection, before a significant degradation of the perceived image quality.

References

[1] W.Bender, D.Gruhl, N.Morimoto and A.Lu, "Techniques for data hiding", *IBM Systems Journal*, **25**, 313-335 (1996).

[2] I. Pitas and T. Kaskalis, Applying Signatures on Digital Images,*Proceedings IEEE Workshop on Nonlinear Signal and Image processing*, Vol I, 460-463 (1995).

[3] J.F.Delaigle, J.M.Boucqueau, J.J.Quisquater and B.Macq, Digital images protection techniques in a broadcast framework : An overview, *Proceedings of ECMAST'96*, vol II, 711-727 (1996).

[4] K.Matsui and K.Tanaka, Video-Steganography: How to secretly embed a signature in a picture, *IMA Intellectual Property Project Proceedings*, vol 1, no 1, (1994).

[5] R.G. van Schyndel, A.Z.Tirkel and C.F.Osborne A Digital Watermark, *Proceedings of ICIP'94*, vol II, 86-89 (1994).

[6] E. Koch and J.Zhao, Towards Robust and Hidden Image Copyright Labeling, *Proceedings IEEE workshop on Nonlinear Signal and Image processing*, Vol I, 452-455 (1995).

[7] C.T.Hsu and J.L.Wu, Hidden signatures in Images, *Proceedings of ICIP'96*, vol III, 223-226 (1996).

[8] I.J.Cox, J.Kilian, T.Leighton and T.Shammoon, Secure Spread spectrum Watermarking for Multimedia, *Tech. Report*, NEC research Institute, no 95-10 (1995).

[9] S.Craver, N.Memon, B.L.Yeo, and M.M.Yeung, Can Invisible Watermarks resolve rightful ownerships?, *IBM Technical Report*, RC 20509 (1996).

[10] N.Nikolaidis and I.Pitas, Robust image watermarking in the spatial domain, to appear in *Signal Proc.* sp. issue on *Copyright Protection and Access control* (1998).

[11] A.G.Bors and I.Pitas, Embedding parametric digital signatures in images, *Proceedings of EUSIPCO'96*, vol.3, 1701-1704 (1996).

[12] M.D.Swanson, B.Zhu, A.H.Tewfik and L.Boney, Robust audio watermarking using perceptual masking, to appear in *Signal Proc.* sp.issue on *Copyright Protection and Access control* (1998).

[13] D.K. Arrowsmith and C.M.Place, *An Introduction to Dynamical systems*, Cambridge Univ. Press (1990).

[14] I. Percival and F.Vivaldi, Arithmetical properties of strongly chaotic systems, *Physica 25D*, 105-130 (1987).

[15] G.Voyatzis and I.Pitas, Chaotic mixing of digital images and applications to watermarking, *Proceedings of ECMAST'96*, vol.2, 687-694 (1996).

[16] N. Jayant, J. Johnston and R. Safranek, Signal Compression Based on Models of Human Perception, *Proceedings of the IEEE*, **81**, no 10, 1385-1422 (1993).

[17] G.Voyatzis and I.Pitas, Applications of Toral automorphisms in image watermarking, *Proceedings of ICIP'96*, vol II, 237-240 (1996).

[18] G. Voyatzis and I. Pitas, Embedding Robust Logo Watermarks in Digital Images, *Proceedings of DSP'97*, vol 1, 213-216 (1997).

[19] A. Papoulis, *"Probability & Statistics"*, Prentice Hall, New Jersey (1991).

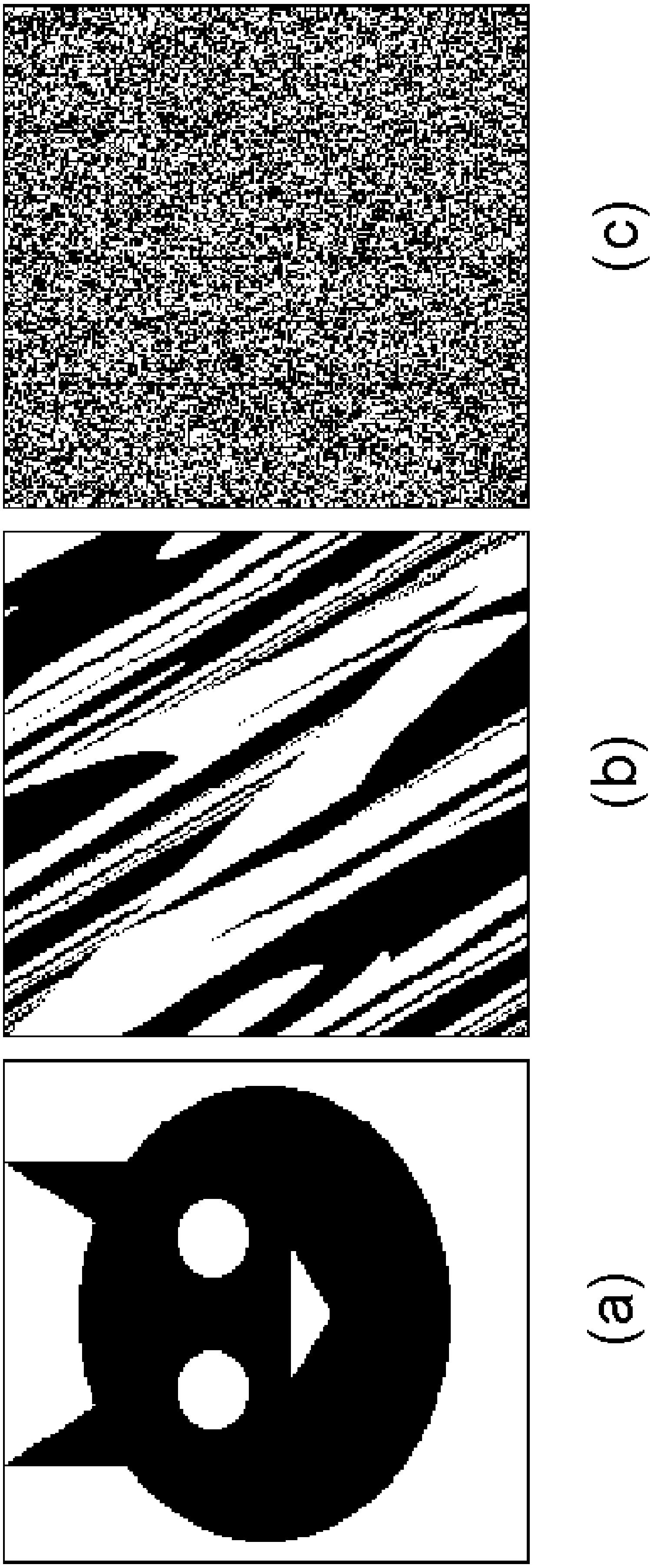

Figure 1: The chaotic mixing of the "cat" (a) for iterations (b) $n = 1$ and (c) $n = 30$.

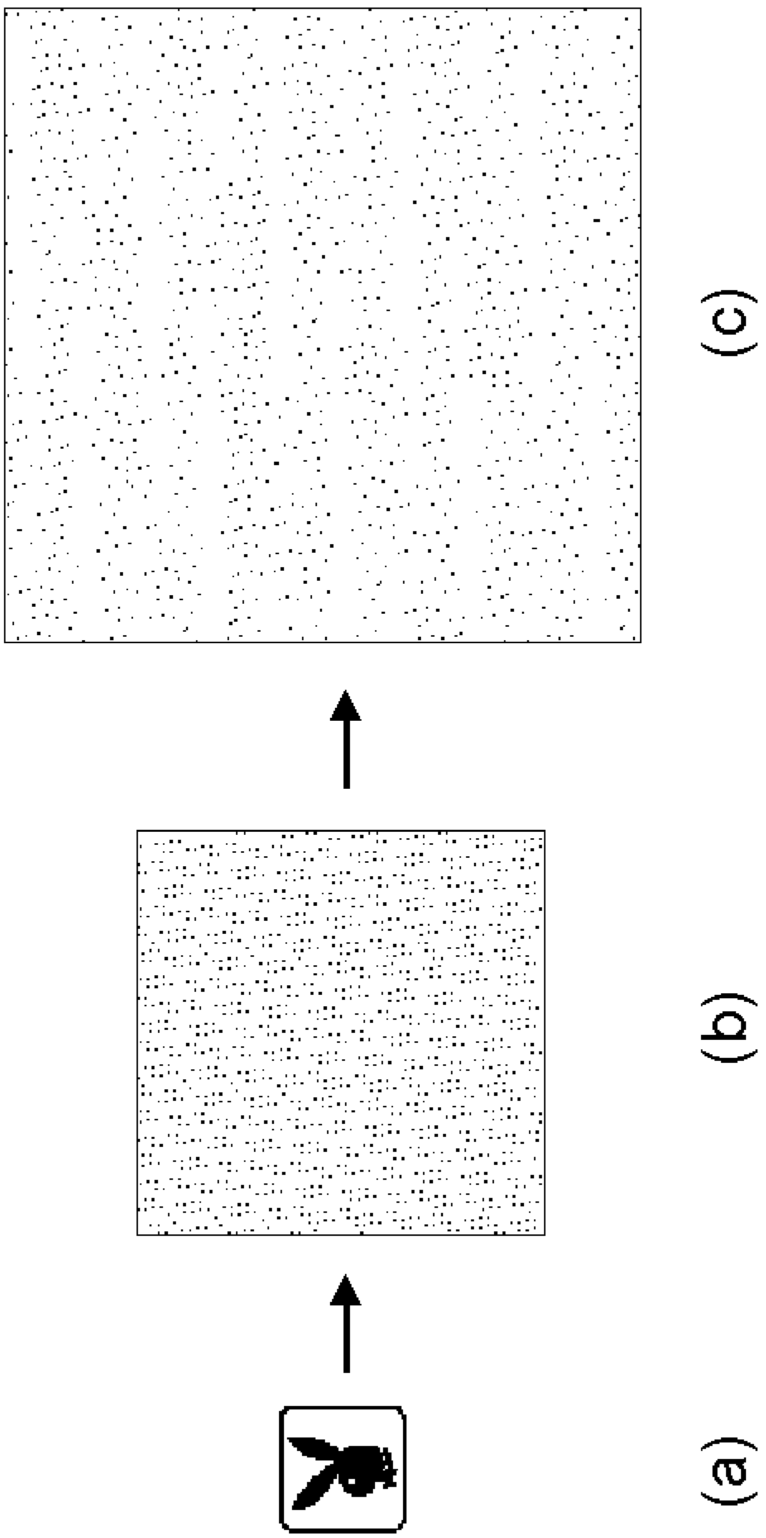

Figure 2: Mixing of the logo "rabbit"(a). Result of (b) first mixing (c) second mixing.

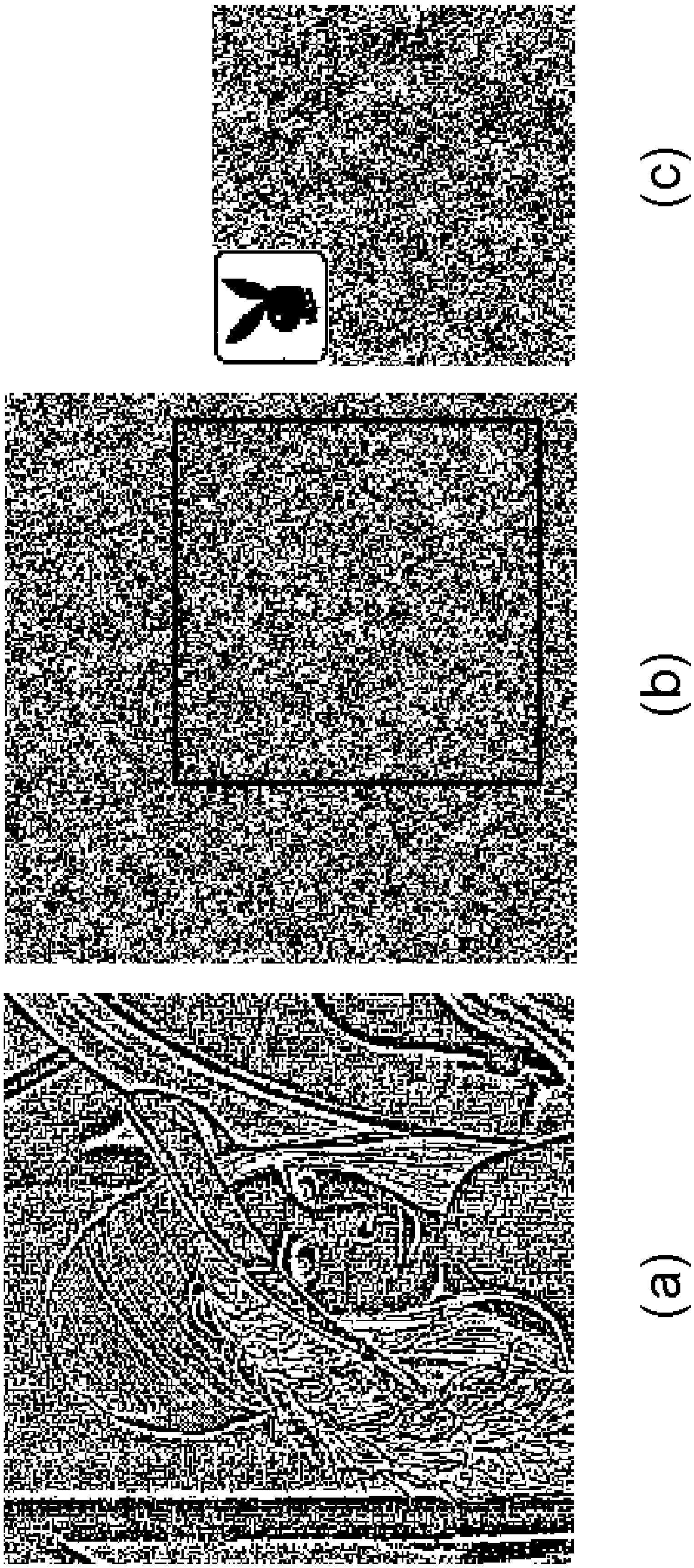

Figure 3: The reconstruction of the logo “rabbit” from a watermarked image. (a) The Z set, (b) result of the first mixing and (c) the watermark reconstruction by mixing the subblock in (b).

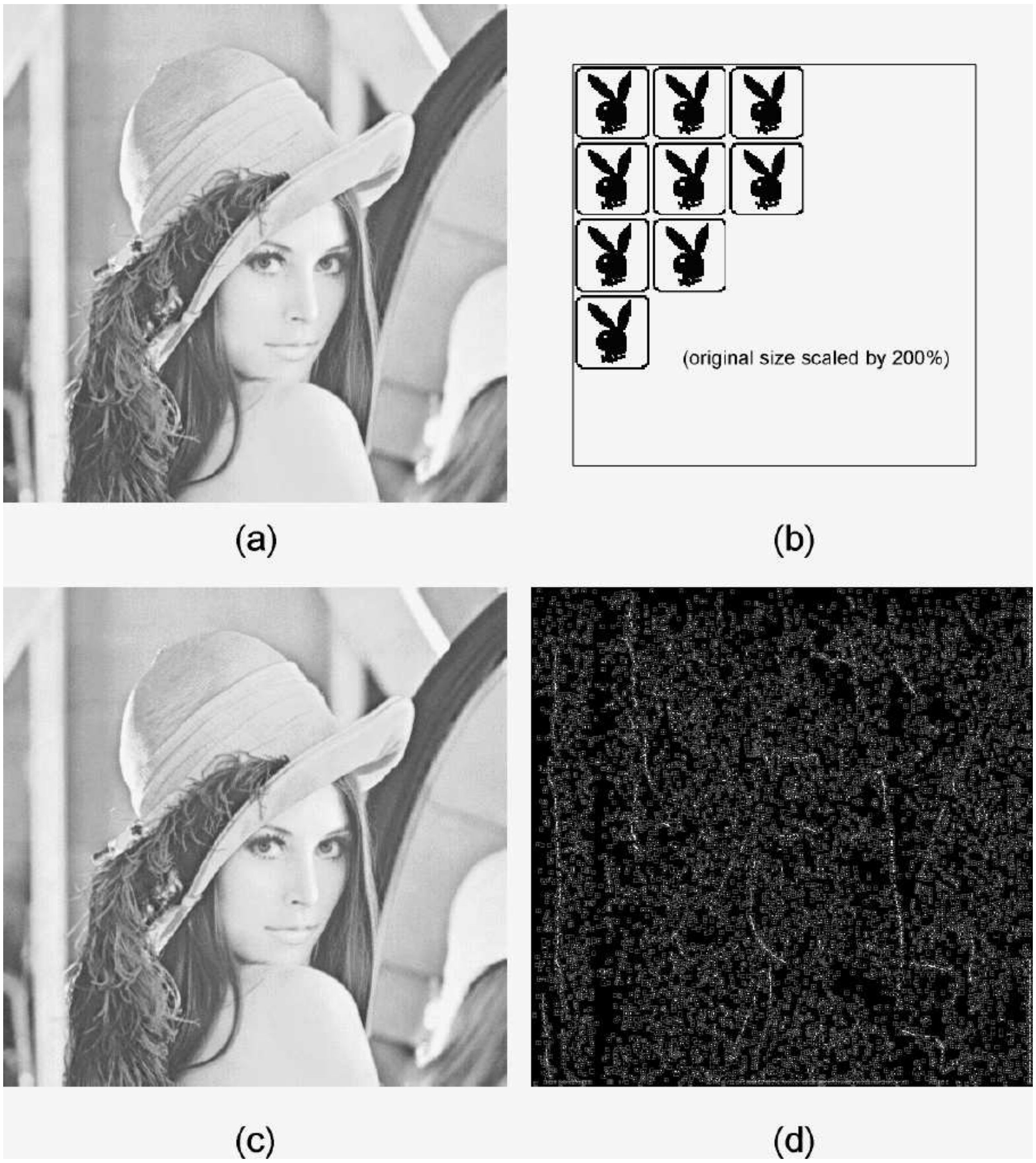

Figure 4: (a) Original image "Lena" of size 512×512, (b) the binary watermark for embedding (c) watermarked "Lena", (d) absolute difference image (intensities have been scaled by 32).

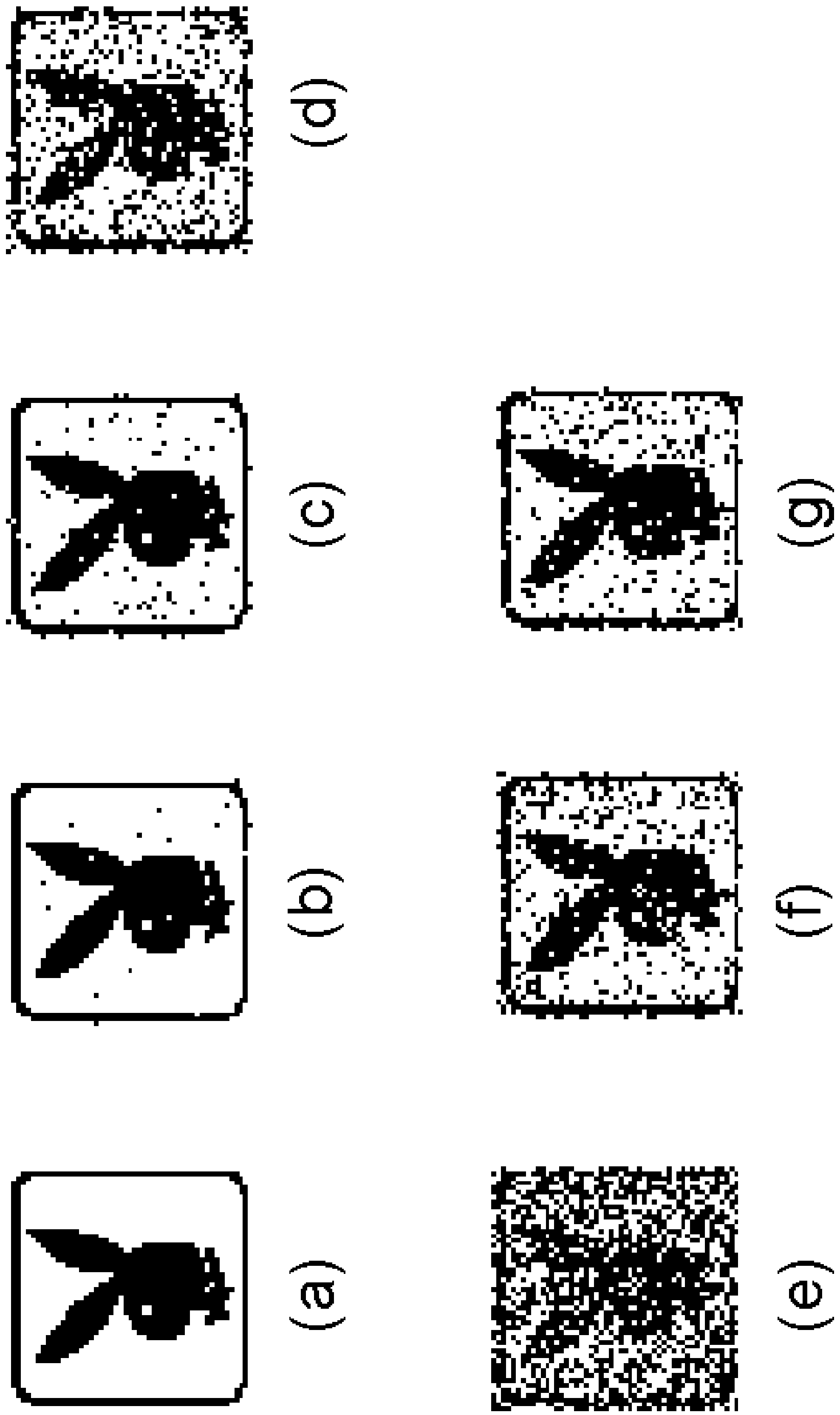

Figure 5: Watermark reconstruction from the watermarked "Lena" modified by (a) Histogram equalization (perfect reconstruction) (b)-(e) JPEG compression 4:1, 6:1, 10:1, and 14:1 respectively (f) 3×3 moving average filter and (g) 3×3 median filter.

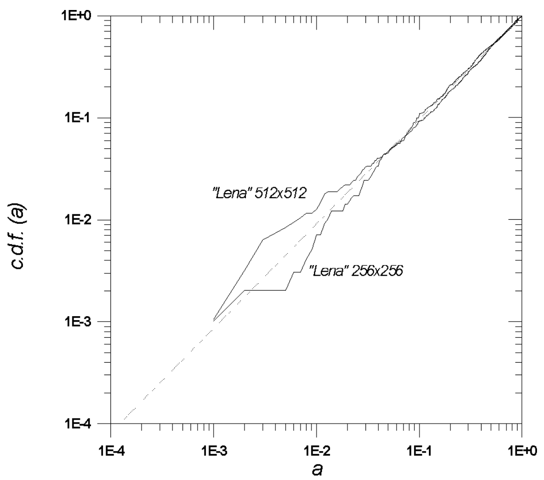

Figure 6: The probability of false alarm error, presented by the *c.d.f.* of the significance a. Dotted line represents the LSQ fitting.

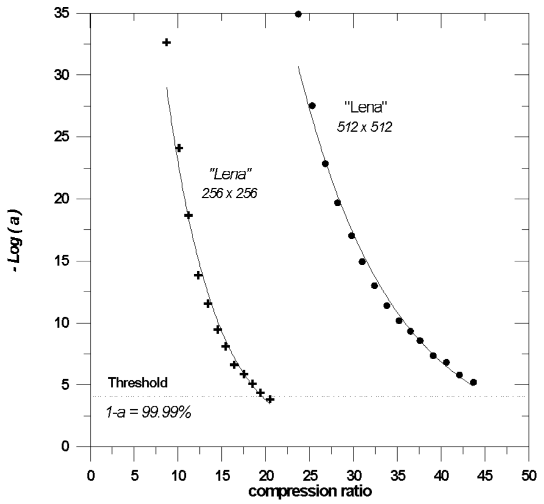

Figure 7: The significance level a versus JPEG compression ratio of the watermarked "Lena". The watermark resists better for the larger image.

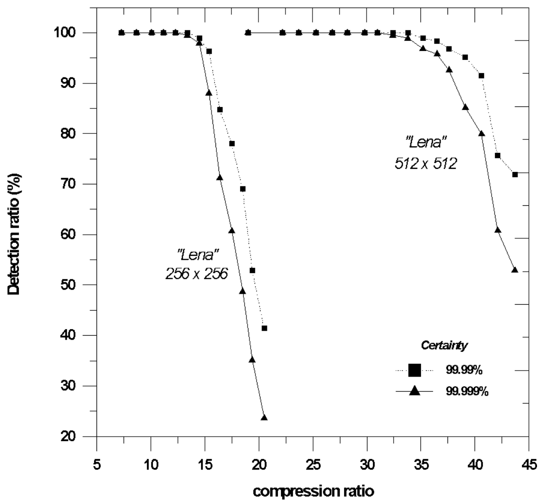

Figure 8: The detection performance for JPEG compressed images.

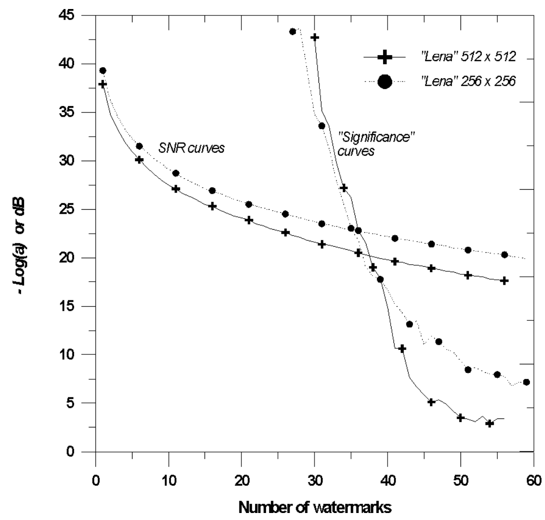

Figure 9: The significance a for the original watermark versus the number of watermarks over-embedded in the same image. The SNR curves indicate the image distortions introduced by the multiple watermarking.