Hindawi Publishing Corporation
EURASIP Journal on Information Security
Volume 2008, Article ID 345184, 14 pages
doi:10.1155/2008/345184

Research Article

Stochastic Image Warping for Improved Watermark Desynchronization

Angela D'Angelo,[1] Mauro Barni,[1] and Neri Merhav[2]

[1] *Department of Information Engineering, University of Siena, 53100 Siena, Italy*
[2] *Department of Electrical Engineering, Technion-Israel Institute of Technology, 32000 Haifa, Israel*

Correspondence should be addressed to Angela D'Angelo, angela.dangelo@unisi.it

Received 28 November 2007; Accepted 19 March 2008

Recommended by Deepa Kundur

The use of digital watermarking in real applications is impeded by the weakness of current available algorithms against signal processing manipulations leading to the desynchronization of the watermark embedder and detector. For this reason, the problem of watermarking under geometric attacks has received considerable attention throughout recent years. Despite their importance, only few classes of geometric attacks are considered in the literature, most of which consist of global geometric attacks. The random bending attack contained in the Stirmark benchmark software is the most popular example of a local geometric transformation. In this paper, we introduce two new classes of local desynchronization attacks (DAs). The effectiveness of the new classes of DAs is evaluated from different perspectives including perceptual intrusiveness and desynchronization efficacy. This can be seen as an initial effort towards the characterization of the whole class of perceptually admissible DAs, a necessary step for the theoretical analysis of the ultimate performance reachable in the presence of watermark desynchronization and for the development of a new class of watermarking algorithms that can efficiently cope with them.

1. INTRODUCTION

Geometric transformations whereby the watermark embedder and detector are desynchronized are known to be one of the most serious threats against any digital watermarking scheme. In the case of still images, for which desynchronization attacks (DAs) can be easily implemented by applying a geometric transformation to the watermarked image, DAs are of the outmost importance, since failing to cope with them would nullify the efficacy of the whole watermarking system.

In the general case, a geometric distortion can be seen as a transformation of the position of the pixels in the image. It is possible to distinguish between global and local geometric distortions. A global transformation is defined by an analytic function that maps the points in the input image to the corresponding points in the output image. It is defined by a set of operational parameters and performed over all the image pixels. Local distortions, instead, refer to transformations affecting in different ways the position of the pixels of the same image or affecting only part of the image. The random bending attack [1], contained in the Stirmark utility, is the most famous example of a local geometric transformation.

Global geometric transformations, especially rotation, scaling, and translation, have been extensively studied in the watermarking literature given their simplicity and diffusion. Though no *perfect* solution exists to cope with geometric attacks, DAs based on global transformations can be handled in a variety of ways, including exhaustive search [2, 3], template-based resynchronization [4–6], self-synchronizing watermarks [7, 8], and watermarking in invariant domains [9]. In all the cases, the proposed solutions rely on the restricted number of parameters specifying the DA. For instance, it is the relatively low cardinality of the set of possible attacks that makes the estimation of the geometric transformation applied by the attacker via exhaustive search or template matching possible (computationally feasible). For this reason, recovering from localized attacks is much harder than recovering from a global attack. A possibility to overcome this problem in case of local attacks could split the search into a number of local searches. However, in this way,

it is likely that the accuracy of the estimation is reduced, given that the estimation would have to rely on a reduced number of samples.

Despite the threats they pose, local geometric transformations have received little attention by the watermarking community. In practice, only the random bending sttack (RBA) contained in the Stirmark software has been studied to some extent. However, even in this case, the real desynchronization capabilities of RBA are not fully understood, given that as implemented in Stirmark, RBA consists of three modules with only one corresponding to a truly local geometric transformation [1].

In this paper, we focus on local geometric attacks for still images. In particular, the aim of our research is twofold:

(i) to introduce two new classes of local DAs that extend the class of local geometric attacks for still images;

(ii) to evaluate the effectiveness of the new attacks and compare them with the classical RBA.

For the above goals, the perceptual impact of the DAs is taken into account since this is the only factor limiting the choice of the attacking strategy. The two models we propose can be seen as a first step towards the characterization of the whole class of perceptually admissible DAs, which in turn is an essential step towards the development of a new class of watermarking systems that can effectively cope with them.

This paper is organized as follows. In Section 2, we describe the RBA contained in the Stirmark software. In Section 3, we introduce a new class of local desynchronization attacks, the LPCD DAs, applied in a full and multiresolution framework. In Section 4, a class of attacks based on Markov random fields is presented. In Section 5, we evaluate the effectiveness of the two new classes of DAs using two simple watermarking systems based on the DCT and DWT transforms. Finally, in Section 6, we summarize the contribution of this work and propose some ideas for future research.

In order to ensure the reproducibility of the experimental results the software, we used for the experiments is available on the web site http://www.dii.unisi.it/~vipp, furthermore a pseudocode description of the algorithms is provided in order to link the software to the global description of the algorithms.

2. STIRMARK RBA

The Stirmark benchmark software first explored RBA's ability to confuse watermark detection. In most of the scientific literature, by RBA, the corresponding geometric attack implemented in the Stirmark software is meant [10], however such an attack is not a truly local attack since it couples three different geometric transformations applied sequentially, only the last of which corresponds to a local attack.

The first transformation applied by Stirmark is defined by

$$\begin{aligned} x' &= t_{10} + t_{11}x + t_{12}y + t_{13}xy, \\ y' &= t_{20} + t_{21}x + t_{22}y + t_{23}xy, \end{aligned} \tag{1}$$

where x', y' are the new coordinates and x, y the old ones. In practice, this transformation corresponds to moving the four corners of the image into four new positions, and modifying coherently all the other sampling positions. The second step is given by

$$\begin{aligned} x'' &= x' + d_{\max} \sin\left(y' \frac{\pi}{M}\right), \\ y'' &= y' + d_{\max} \sin\left(x' \frac{\pi}{N}\right), \end{aligned} \tag{2}$$

where M and N are the vertical and horizontal dimensions of the image. This transformation applies a displacement which is zero at the border of the image and maximum ($d_{\max}$) in the center. The third step of the Stirmark geometric attack is expressed as

$$\begin{aligned} x''' &= x'' + \delta_{\max} \sin(2\pi f_x x'') \sin(2\pi f_y y'') \operatorname{rand}_x(x'', y''), \\ y''' &= y'' + \delta_{\max} \sin(2\pi f_x x'') \sin(2\pi f_y y'') \operatorname{rand}_y(x'', y''), \end{aligned} \tag{3}$$

where f_x and f_y are two frequencies (usually smaller than 1/20) that depend on the image size, and $\operatorname{rand}_x(x'', y'')$ and $\operatorname{rand}_y(x'', y'')$ are random numbers in the interval $[1, 2)$. However, (3) is the only local component of the Stirmark attack since it introduces a random displacement at every pixel position. In the sequel by RBA, we will mean only the transformation expressed by (3). This can be obtained by using the Stirmark software setting to 0 the b, d, i, and o parameters (resp., the bending factor, the maximum variation of a pixel value, the maximum distance a corner can move inwards and outwards), and leaving R (the randomisation factor) to the default value of 0.1.

3. THE CLASS OF LPCD DAS

In this section, we describe a first new class of DAs, namely, local permutation with cancelation and duplication (LPCD) DAs. We start from the plain LPCD attack, then we pass to the C-LPCD (constrained LPCD). Finally, we consider the multiresolution extension of the above two classes.

3.1. LPCD

By focusing on the 1D case, let $y = \{y(1), \text{let } y(2), \ldots, y(n)\}$ be a generic signal, and let $z = \{z(1), z(2), \ldots, z(n)\}$ be the distorted version of y. The LPCD model states that $z(i) = y(i + \Delta_i)$, where Δ_i is a sequences of i.i.d random variables uniformly distributed in a predefined interval $I = [-\Delta, \Delta]$. For simplicity, we assume that Δ_i can take only integer values in I. This way, the values assumed by the samples of z are chosen among those of y. The above model yields an interesting interpretation of the attacked signal z. To introduce it, it is convenient to describe the LPCD attack as a channel $W(z \mid y)$ defined as follows (neglecting edge effects):

$$W(z \mid y) = \prod_{i=1}^{n} W(z(i) \mid y_{i-\Delta}^{i+\Delta}), \tag{4}$$

where (i) y_i^j, for $i \le j$, denotes $(y(i), y(i+1), \ldots, y(j))$ (a similar notation convention applies to z), and (ii)

$$W(z(i) \mid y_{i-\Delta}^{i+\Delta}) = \frac{1}{2\Delta+1} \sum_{k=-\Delta}^{\Delta} \mathbf{1}\{z(i) = y(i-k)\}, \quad (5)$$

where $\mathbf{1}\{z(i) = y(i-k)\}$ denotes the indicator function of the event $\{z(i) = y(i-k)\}$. According to the above equation, the LPCD channel $W(z(i) \mid y_{i-\Delta}^{i+\Delta})$ assigns the same probability, $1/(2\Delta+1)$, and independently, to all possible values of $k \in \{-\Delta, -\Delta+1, \ldots, \Delta\}$, and it picks $z(i) = y(i-k)$. However, *any* other probability assignment $W(z(i) \mid y_{i-\Delta}^{i+\Delta})$ is allowed. Likewise, the probability law of y does not need to be known (except the fact that it is memoryless). An equivalent representation of this model is obtained by defining $u(i) = y_{i-\Delta}^{i+\Delta}$. Here, if $\{y(i)\}$ are i.i.d., then $\{u(i)\}$ is a first-order Markov process. Also, the channel W from $u = (u(1), \ldots, u(n))$ to y is obviously memoryless according to (4). Thus, z is governed by a hidden Markov process:

$$Q(z) = \sum_{u} \prod_{i=1}^{n} [P(u(i) \mid u(i-1)) W(z(i) \mid u(i))]. \quad (6)$$

The above interpretation of the LPCD model may open the way to the definition of optimum embedding and detection strategies along the same lines described in [11].

To extend the 1D-LPCD model to the two-dimensional case, if $Z(i, j)$ is a generic pixel of the distorted image Z, we let

$$Z(i, j) = Y(i + \Delta_h(i, j), j + \Delta_v(i, j)), \quad (7)$$

where Y is the original image and $\Delta_h(i, j)$ and $\Delta_v(i, j)$ are i.i.d. integer random variables uniformly distributed in the interval $[-\Delta, \Delta]$.

3.2. C-LPCD

An important limitation of the LPCD model is the lack of memory. This is likely to be a problem from a perceptual point of view: with no constraints on the smoothness of the displacement field, there is no guarantee that the set of LPCD distortions is perceptually admissible even by considering very small values of Δ.

One way to overcome the limitation of the LPCD model, and to obtain better results from a perceptual point of view, is to require that the sample order, in the 1D case, is preserved (thus introducing memory in the system). In practice, the displacement of each element i of the distorted sequence z is conditioned on the displacement of the element $i-1$ of the same sequence. In formulas, $z(i) = y(i + \Delta_i)$, where Δ_i is a sequence of i.i.d integer random variables uniformly distributed in the interval $I = [\max(-\Delta, \Delta_{i-1} - 1), \Delta]$. In the sequel, we will refer to this new class of DAs as C-LPCD (constrained local permutation with cancelation and duplication). Figure 1 illustrates the behavior of the C-LPCD model in the 1D case with $\Delta = 2$. We know that $z(i) = y(i + \Delta_i)$, let us assume that Δ_i is chosen in the interval $I_i = [-2, 2]$ (the solid-line box) and that $\Delta_i = 1$, it means

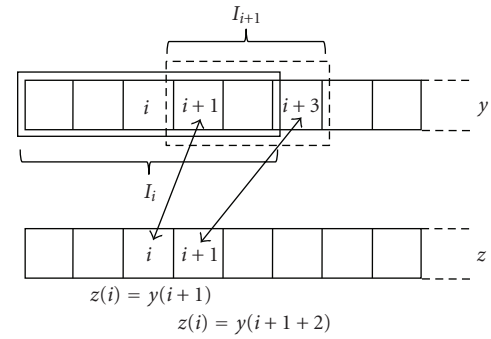


Figure 1: Constrained LPCD with $\Delta = 2$ (one-dimensional case).

that $z(i) = y(i+1)$. At the next step, we know that $z(i+1) = y(i+1+\Delta_{i+1})$, where Δ_{i+1}, due to the position of the pixel $z(i)$, must be chosen in the interval $I_{i+1} = [0, 2]$ (the bold dotted-line box). The interval I_{i+1} is smaller than I_i because the position of the element $i+1$ cannot precede that of the element i. For example, Δ_{i+1} could be equal to 2 yielding $z(i+1) = y(i+3)$.

The C-LPCD model can be mathematically described by resorting to the theory of Markov chains. For simplicity, let us focus again on the one-dimensional case. It is possible to design a Markov chain whose states correspond to the possible sizes of the interval $I = [\max(-\Delta, \Delta_{i-1} - 1), \Delta]$.

In a general case, given Δ, the maximum size of I is equal to $N = 2\Delta + 1$ (the minimum being equal to 2) and the transition matrix of the Markov chain (whose size is $2\Delta \times 2\Delta$) is

$$P = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & \cdots & \cdots & \cdots & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & \cdots & \cdots & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{1}{2\Delta+1} & \frac{1}{2\Delta+1} & \frac{1}{2\Delta+1} & \cdots & \cdots & \cdots & \frac{2}{2\Delta+1} \end{bmatrix}, \quad (8)$$

where each element p_{ij} of the matrix is the transition probability of going from state i to state j.

A visual inspection conducted on a set of images distorted with the C-LPCD model reveals that changing the value of Δ does not change the perceived intensity of the deformation.

This effect, which can be described by resorting to the properties of Markov chains [12], can be avoided by allowing the model to generate a larger variety of displacement fields. For this reason, we modified the Markov chain by changing the transition probabilities among the states in order to give a greater probability to the transitions that result in a larger interval I. A way to do this is to assign the same probability (equal to $1/(2\Delta+1)$) to the transitions that cause a decrease of the size of I, corresponding to the elements i, j with $i = 1, \ldots, \Delta$ and $j = 1, \ldots, i$ of the transition matrix,

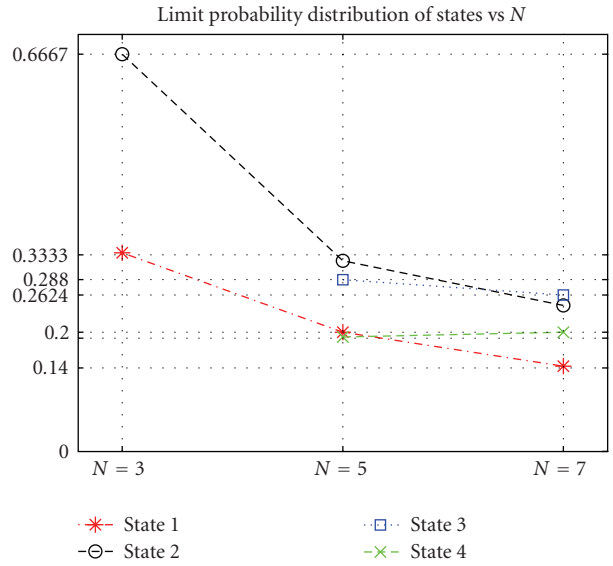


Figure 2: Limit probability distribution of states versus Δ ($N = 2\Delta + 1$).

and to assign all the remaining probability mass, equal to $1 - \sum_{j=1}^{i} p_{ij}$, to the transition corresponding to the element i, j with $i = 1, \ldots, \Delta$ and $j = i + 1$, that is, the transition whose effect is to enlarge the interval I. The corresponding transition matrix becomes

$$P = \begin{bmatrix} \frac{1}{2\Delta+1} & \frac{2\Delta}{2\Delta+1} & 0 & \cdots & \cdots & \cdots & 0 \\ \frac{1}{2\Delta+1} & \frac{1}{2\Delta+1} & \frac{2\Delta-1}{2\Delta+1} & 0 & \cdots & \cdots & 0 \\ \frac{1}{2\Delta+1} & \frac{1}{2\Delta+1} & \frac{1}{2\Delta+1} & \frac{2\Delta-2}{2\Delta+1} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{1}{2\Delta+1} & \frac{1}{2\Delta+1} & \frac{1}{2\Delta+1} & \cdots & \cdots & \cdots & \frac{2}{2\Delta+1} \end{bmatrix}. \tag{9}$$

Figure 2 shows the limit probability distribution of the states versus Δ for the described Markov chain.

By looking at the figure, it is possible to note that regardless of the value of Δ, all the states have almost the same limit probabilities.

The extension of the C-LPCD model to the 2D case is obtained by applying the 1D algorithm by rows to obtain the horizontal displacement field $\Delta_h(i, j)$, and by columns for the vertical displacements $\Delta_v(i, j)$.

3.3. Multiresolution extension

To make the distortion less perceptible, we considered a multiresolution version of the LPCD and C-LPCD attacks, whereby the DAs are applied at different resolutions to obtain the global displacement field: a low-resolution displacement field is first generated, then a full-size displacement is built by means of a bicubic interpolation. The full resolution field is applied to the original image to produce the distorted image.

More specifically, the multiresolution models consist of two steps. Let $S \times S$ be the size of the image (for sake of simplicity, we assume that S is a power of 2). To apply the LPCD (or C-LPCD) model at the Lth level of resolution, two displacement fields $\delta_h(i, j)$ and $\delta_v(i, j)$ with size $S/2^L \times S/2^L$ are generated. Then, the full-resolution fields $\Delta_h(i, j)$ and $\Delta_v(i, j)$ are built by means of bicubic interpolation. Note that this way noninteger displacement values are introduced. It is still possible to obtain integer displacements by applying a nearest neighbor interpolation instead of a bicubic one (of course at the expense of the smoothness of the displacement field). The full-resolution displacement fields Δ_h and Δ_v are used to generate the warped image Z as follows:

$$Z(i, j) = Y(i + \Delta_h(i, j), j + \Delta_v(i, j)). \tag{10}$$

As opposed to the original version of LPCD and C-LPCD, however, the presence of noninteger displacements is now possible due to the interpolation. To account for this possibility, whenever the displacement vector points to noninteger coordinates of the original image, the gray level of the attacked image $Z(i, j)$ is computed by means of the bicubic interpolation. While the above interpolation does not have a significant impact on the visual quality of the attacked image, the possible introduction of new gray levels, which were not present in the original image, complicates the LPCD and C-LPCD models, by making it more difficult to describe the attacked signal as a hidden Markov process (as we did in Section 3.1).

The pseudocode description of the multiresolution version of LPCD DAs is provided by Algorithms 1 and 2.

3.4. Cardinality evaluation

A measure of the difficulty of coping with a given type of DA is given by the cardinality of the attack class. In fact, the larger is the DA space, the more difficult will be to recover the synchronization between the embedded and the detector, both in terms of complexity and accuracy. As a matter of fact, it is possible to show [3, 11] that as long as the cardinality of the DAs is subexponential, the exhaustive search of the watermark results in asymptotically optimum watermark detection with no loss of accuracy with regard to false-detection probability. By contrast, when the size of the DA is exponential, simply considering all the possible distortions may not be a feasible solution both from the point of view of computational complexity and detection accuracy [11]. In order to evaluate the cardinality of the classes of DAs, the perceptual impact of LPCD and C-LPCD must be taken into account. Thus, we first found the limits of the model parameters by means of perceptual considerations, then we estimated the cardinality of the various classes of LPCD DAs.

Let us observe that from a perceptual point of view, LPCD DAs have a different behavior for different values of N and for different levels of resolution L, in particular, the image quality increases if the attacks are applied to a lower level of resolution (larger L) but, at the same time, the number of possible distortions decreases.

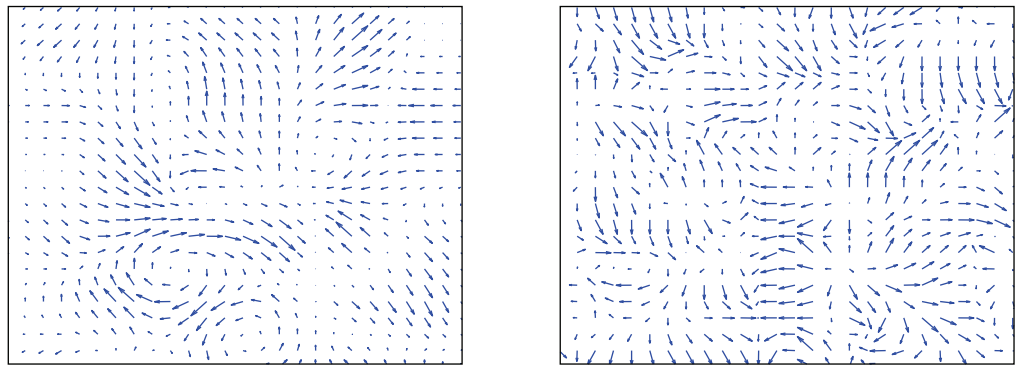
In a previous work [13], both subjective and objective tests were performed to establish the sensitivity of the human

1. Read image to be attacked Y, read size of the window Δ, read level of resolution L
2. dim = size(image)/2^L {size of the low resolution displacement field}
3. Initialize matrices δ_h and δ_v of horizontal and vertical displacement fields to 0
4. *for* $i = 1 : \text{dim}$ *do*
5. *for* $j = 1 : \text{dim}$ *do*
6. If $(i < \Delta + 1)$ or $(j < \Delta + 1)$ *then*
7. $\delta_h(i, j)$ and $\delta_v(i, j)$ are randomly chosen in $[-(\min(i, j) - 1); (\min(i, j) - 1)]$
8. *else if* $(i > \text{dim} - \Delta)$ or $(j > \text{dim} - \Delta)$ *then*
9. $\delta_h(i, j)$ and $\delta_v(i, j)$ are randomly chosen in $[-(\text{dim} - \max(i, j)); (\text{dim} - \max(i, j))]$
10. *else*
11. $\delta_h(i, j)$ and $\delta_v(i, j)$ are randomly chosen in $[-\Delta; \Delta]$
12. *end if*
13. *end for*
14. *end for*
15. Resize the displacement fields given by δ_h and δ_v to the image size through bicubic interpolation provided by the matlab function *imresize*{to obtain the high resolution displacement fields Δ_h an Δ_v}
16. *for* $i = 1 : \text{size(image)}$ *do*
17. *for* $j = 1 : \text{size(image)}$ *do*
18. $Z(i, j) = Y(i + \Delta_h(i, j), j + \Delta_v(i, j))$ {Apply the displacement fields to the image, to obtain the attacked image Z, by means of bicubic interpolation}
19. *end for*
20. *end for*

Algorithm 1: LPCD model.

1. Read image to be attacked, read size of the window Δ, read level of resolution L
2. dim = size(image)/2^L {size of the low resolution displacement field}
3. Initialize matrices δ_h and δ_v of horizontal and vertical displacement fields to 0
4. *for* $i = 1 : \text{dim}$ *do*
5. *for* $j = 1 : \text{dim}$ *do*
6. *if* $(i < \Delta + 1)$ or $(j < \Delta + 1)$ *then*
7. $\delta_h(i, j)$ and $\delta_v(i, j)$ are randomly chosen in $[-(\min(i, j) - 1); (\min(i, j) - 1)]$
8. *else if* $(i > \text{dim} - \Delta)$ or $(j > \text{dim} - \Delta)$ *Then*
9. $\delta_h(i, j)$ and $\delta_v(i, j)$ are randomly chosen in $[-(\text{dim} - \max(i, j)); (\text{dim} - \max(i, j))]$
10. *else*
11. $\delta_h(i, j)$ is chosen in $Ix = [\max(\Delta, \delta_h(i - 1, j) - 1), \Delta]$ with a distribution vector $P = [1 - (\text{size}(Ix) - 1)/\Delta; 1/\Delta; \ldots; 1/\Delta]$
12. $\delta_v(i, j)$ is chosen in $Iy = [\max(\Delta, \delta_v(i - 1, j) - 1), \Delta]$ with a distribution vector $P = [1 - (\text{size}(Iy) - 1)/\Delta; 1/\Delta; \ldots; 1/\Delta]$
13. *end if*
14. *end for*
15. *end for*
16. Resize the displacement fields given by δ_h and δ_v to the image size through bicubic interpolation provided by the matlab function *imresize*{to obtain the high resolution displacement fields Δ_h an Δ_v}
17. *for* $i = 1 : \text{size(image)}$ *do*
18. *for* $j = 1 : \text{size(image)}$ *do*
19. $Z(i, j) = Y(i + \Delta_h(i, j), j + \Delta_v(i, j))$ {Apply the displacement fields to the image, to obtain the attacked image Z, by means of bicubic interpolation}
20. *end for*
21. *end for*

Algorithm 2: Constrained LPCD model (modified version).



(a) (b)

Figure 3: Examples of displacement fields generated with LPCD DA's: (a) LPCD with $L = 6$ and $N = 5$; (b) C-LPCD with $L = 5$ and $N = 5$.

visual system to the geometric distortions introduced by the LPCD model as a function of the control parameters N and L. This way, the authors were able to identify the range of values of the control parameters that do not affect image quality: for each level of resolution, the maximum value of N that can be used while keeping the distortion invisible was found. For instance, in the case of images of size 512×512, the maximum admissible geometric distortions are obtained by using $L = 6$, $N = 5$ for the LPCD model and $L = 5$, $N = 5$ or $L = 6$, $N = 7$ for the C-LPCD model (for higher level of resolution, it is not possible to find an adequate value of N resulting in an invisible distortion).

In Figure 3, two examples of displacement fields generated with the LPCD attack with $L = 6$ and $N = 5$ Figure 3(a) and the C-LPCD attack with $L = 5$ and $N = 5$ Figure 3(b) are given: as expected, by applying the model to a lower level of resolution, it is possible to obtain a more uniform field (for the purpose of visibility, the total displacement field is cropped and only one vector every sixteen samples is depicted in the figure).

We can now use the above considerations to estimate the cardinality of the class of LPCD DAs. For the LPCD model, the number of possible admissible geometric distortions is simply equal, neglecting the boundary effects, to $(N^{S/2^L \times S/2^L})^2$, where S is the size of the image. Then, if we consider a 512×512 image, and if we take into account the perceptual analysis in [13], then we obtain 2.93×10^{89} different attacked images.

With regard to the C-LPCD model, we need to refer again to the theory of Markov chains. Let us consider the one-dimensional case and the graph of the Markov chain describing the C-LPCD model. It is possible to construct the adjacency matrix A of zeroes and ones, where $A_{i,j} = 1$ if in the graph there is an edge going from node i to node j and zero, otherwise. The number of paths of length n that start from node i and end into node j is given by the (i, j) entry of the matrix A^n. The exponential growth rate of the number of paths of length n in the graph is $e^{n \ln \lambda_{\max}}$, where $\lambda_{\max}$ is the largest eigenvalue of A. In the C-LPCD case, the practical values of n are not very large, for instance, for a 512×512 image, with $L = 5$, we have $n = 16$, then we can easily compute the matrix A^n and derive the exact size of the C-LPCD class of attacks. Specifically, by remembering that the two-dimensional extension of C-LPCD is obtained by applying the one-dimensional C-LPCD DA first by rows and then by columns, we obtain the results reported in Table 1.

Table 1: Cardinality evaluation of the LPCD attacks: in the first row, the number of possible distortions is reported, the second row refers to the number of typical sequences.

	LPCD $L6 - N5$	C-LPCD $L5 - N5$	C-LPCD $L6 - N7$
Cardinality	2.93×10^{89}	1.54×10^{265}	1.54×10^{84}
2^{nH}	2.93×10^{89}	4.76×10^{114}	8.53×10^{30}

With the above approach, we were able to count all the distortions that can be generated with the C-LPCD model. Nevertheless, as explained in the previous subsection, the occurrence of a particular distortion configuration depends on the Markov chain-transition matrix and is not constant for all the configurations. Thus, for a more appropriate evaluation of the cardinality of C-LPCD DAs, we need to refer to the entropy rate of the corresponding Markov chain. In this context, the following result from information theory [14] is useful: let $\{X_i\}$ be a stationary Markov chain with stationary distribution μ and transition matrix P, then the entropy rate is

$$H(\mathcal{X}) = -\sum_{ij} \mu_i p_{ij} \log p_{ij}. \tag{11}$$

The knowledge of the entropy rate of the Markov chain and the asymptotic equipartition property (AEP) [14] help us to find the number of possible distortions that can be generated with a so-defined Markov chain, since it asymptotically corresponds to the number of typical sequences, that is, 2^{nH}. After some algebraic manipulations, we find that in the case of C-LPCD with $N = 5$ and $L = 5$, $H(\mathcal{X})$ is approximately equal to 1.4881 bits and the number of different distortions that is possible to generate is $2^{256 \cdot 1.4881} \simeq 4.76 \cdot 10^{114}$. In the same way, in the case of C-LPCD with $N = 7$ and $L = 6$, it is possible to generate $2^{64 \cdot 1.6055} \simeq 8.53 \cdot 10^{30}$ different distortions. By looking at Table 1, we can see that,

as we expected, the cardinality of C-LPCD evaluated by considering the entropy rate of the Markov chain (second row) is much smaller than the number of possible distortions (first row). We conclude this section by observing that the size of both the LPCD and the C-LPCD DAs exhibit an exponential growth, with the constrained model resulting in a higher growth rate. For this reason, both classes of attacks are likely to make watermark detection rather difficult, and will need to be carefully considered in future works on DA-resistent watermarking.

4. MARKOV FIELD DA (MF-DA)

One problem with the C-LPCD attack is that it does not take into account the two-dimensional nature of images since it is based on a one-dimensional Markov chain. To overcome this limitation, we introduce a new class of DAs based on the theory of Markov random fields. We will refer to this new class of attacks as MF-DA.

Markov random field theory is a branch of probability theory for analyzing the spatial or contextual dependencies of physical phenomena. The foundations of the theory of Markov random fields may be found in statistical physics of magnetic materials (Ising models, spin glasses, etc.) and also in solids and crystals, where the molecules are arranged in a lattice structure and there are interactions with close neighbors (e.g., Debye's theory for the vibration of atoms in a lattice is based on a model of quantum harmonic oscillators with coupling among nearest neighbors). Markov random fields are often used in image processing applications, because this approach defines a model for describing the correlation among neighboring pixels [15].

4.1. Model description

Many vision problems can be posed as labeling problems in which the solution of a problem is a set of labels assigned to image pixels or features. A labeling problem is specified in terms of a set of sites and a set of labels. Let $\mathcal{S} = \{1, \ldots, m\}$ be a discrete set of m sites in which $1, \ldots, m$ are indices (a site often represents a point or a region in the Euclidean space such as an image pixel or an image feature). A label is an event that may happen to a site. Let $\mathcal{L} = \{l_1, \ldots, l_n\}$ be a set of labels. The labeling problem is to assign a label from $\mathcal{L}$ to each of the sites in $\mathcal{S}$. In the terminology of random fields, a labeling is called a configuration.

The sites in $\mathcal{S}$ are related to one another via a neighborhood system. A neighborhood system for $\mathcal{S}$ is defined as $N = \{N_i \mid \forall i \in \mathcal{S}\}$, where N_i is the set of sites neighboring i. The neighboring relationship has the following properties:

(1) a site is not neighboring to itself: $i \notin N_i$,

(2) the neighboring relationship is mutual: $i \in N_{i'} \Leftrightarrow i' \in N_i$.

If $\mathcal{S}$ is a regular lattice, the neighboring set of i is often defined as the set of nearby sites within a radius of r:

$$N_i = \{i' \in \mathcal{S} \mid [\mathrm{dist}(i, i')]^2 \leq r,\ i' \neq i\}. \tag{12}$$

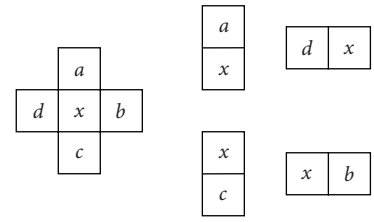


Figure 4: Structure of a first-order neighborhood system and corresponding pair-sites cliques.

Once introduced a set $\mathcal{S}$ and a neighborhood system N, it is possible to define a clique c for $(\mathcal{S}, N)$ like a subset of sites in $\mathcal{S}$. It consists either of a single-site $c = \{i\}$ (single-site clique), or a pair of neighboring sites $c = \{i, i'\}$ (pair-sites cliques), or a triple of neighboring sites $c = \{i, i', i''\}$ (triple-sites cliques), and so on.

The collections of single-site, pair-site, and triple-site, cliques will be denoted by $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$, respectively, where

$$\begin{aligned} \mathcal{C}_1 &= \{i \mid \in \mathcal{S}\}, \\ \mathcal{C}_2 &= \{\{i, i'\} \mid i' \in \mathcal{N}_i,\ i \in \mathcal{S}\}, \\ \mathcal{C}_3 &= \{\{i, i', i''\} \mid i, i', i'' \in \mathcal{S} \text{ are neighbors to one another}\}. \end{aligned} \tag{13}$$

The collection of all cliques for $(\mathcal{S}, N)$ is denoted by $\mathcal{C}$.

Figure 4 shows a first-order neighborhood system, also called a 4 -neighborhood system, with the four corresponding pair-sites cliques. The **x** symbol denotes the considered site and the letters indicate its neighbors.

A random field $F = \{F_1, F_2, \ldots, F_m\}$ is a family of random variables defined on a set $\mathcal{S}$, in which each random variable F_i takes a value f_i in a set of labels $\mathcal{L}$.

F is said to be a Markov random field (MRF) on $\mathcal{S}$ with respect to a neighborhood system N if and only if the two following conditions are satisfied:

$$\begin{aligned} &P(f) > 0, \quad \forall f \in \mathcal{L}^m \text{ (positivity)}, \\ &P(f_i \mid f_{S-\{i\}}) = P(f_i \mid f_{N_i}), \quad \forall i \in \mathcal{S} \text{ (Markov property)}, \end{aligned} \tag{14}$$

where $f = \{f_1, \ldots, f_m\}$ is a configuration of F (corresponding to a realization of the field), $P(f)$ is the joint probability $P(F_1 = f_1, \ldots, F_m = f_m)$ of the joint event $F = f$, that is, it measures the probability of the occurrence of a particular configuration, and

$$f_{N_i} = \{f_{i'}, i' \in N_i\} \tag{15}$$

denotes the set of values at the sites neighboring i, that is, the neighborhood N centered at position i. The positivity is due to technical reasons, since it is a necessary condition if we want the Hammersley-Clifford theorem (see below) to hold [16].

To exploit MRFs characteristics in a practical way, we need to refer to the Hammersley-Clifford theorem [15] for which F is an MRF on $\mathcal{S}$ with respect to N if and only if F is a Gibbs random field (GRF) on $\mathcal{S}$ with respect to N, that

is, the probability distribution of an MRF has the form of a Gibbs distribution:

$$P(f) = \frac{e^{-(1/T)U(f)}}{Z}, \tag{16}$$

where Z is a normalizing constant called the partition function, T is a constant called the temperature, and $U(f)$ is the energy function. The energy function

$$U(f) = \sum_{c \in \mathcal{C}} V_c(f) \tag{17}$$

is a sum of cliques potentials, $V_c(f)$, over all possible cliques $\mathcal{C}$. Thus the value of $V_c(f)$ depends on the local configuration on the clique c. The practical value of the theorem is that it provides a simple way of specifying the joint probability. Since $P(f)$ measures the probability of the occurrence of a particular configuration, we know that the more probable configurations are those with lower energies.

In our case, we can model geometric attacks with a random field F defined on the set S of the image pixels. The value assumed by each random variable represents the displacement associated to a particular pixel. Specifically, for each pixel, we have two values for the two directions x and y. For this reason, each variable F_i is assigned a displacement vector $\mathbf{f}_i = (f_x, f_y) \in \mathcal{L} \times \mathcal{L}$. The advantage brought by MRF theory is that by letting the displacement field of a generic point (x, y) of the image depend on the displacement fields of the other points of its neighborhood (let us indicate this set with the notation $N(x, y)$), we can automatically impose that the resulting displacement field is smooth enough to avoid annoying geometrical distortions.

As we said, an MRF is uniquely determined once the Gibbs distribution and the neighborhood system are defined. In the approach proposed here, for each pixel (x, y), only four neighbors of first order and the corresponding four pair-site cliques, as described by Figure 4. The potential function we used is a bivariate normal distribution expressed by:

$$V_{((x,y),(\tilde{x},\tilde{y}))} = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left\{ -\left[\frac{(f_x - f_{\tilde{x}})^2}{2\sigma_x^2} + \frac{(f_y - f_{\tilde{y}})^2}{2\sigma_y^2} \right] \right\}, \tag{18}$$

where f_x and f_y are the components of the displacement vector $\mathbf{f}_{(x,y)}$ associated to the pixel (x, y), $(\tilde{x}, \tilde{y})$ is a point belonging to the 4-neighborhood of (x, y), $f_{\tilde{x}}$ and $f_{\tilde{y}}$ are the x,y components of the displacement vector $\mathbf{f}_{(\tilde{x},\tilde{y})}$ associated to the pixel $(\tilde{x}, \tilde{y})$ and σ_x and σ_y are the two components of the standard deviation vector σ (these values are controlled by perceptual constraints).

A typical application of MRF in the image processing field is to recover the original version of an image (or a motion vector field) by relying on a noisy version of the image. By assuming that the original image can be described by means of an MRF, the above problem is formulated as a maximum a posteriori estimation problem. Thanks to the Hammersley-Clifford theorem, this corresponds to an energy minimization problem that is usually solved by applying an iterative relaxation algorithm to the noisy version of the image [16]. The problem we have to face here, however, is slightly different. We simply want to generate a displacement field according to the Gibbs probability distribution defined by (16) and the particular potential function expressed in (18).

To do so, the displacement field is initialized by assigning to each pixel (x, y) in the image a displacement vector $\mathbf{f}_{(x,y)}$ generated randomly (and independently on the other pixels) in the interval in $\mathcal{L} \times \mathcal{L}$ with $\mathcal{L} = \{f \in \mathbb{Z} : -c \leq f \leq c\}$ (the value of c is determined by relying on perceptual considerations). This initial random field is treated as a noisy version of an underlying displacement field obeying the MF-DA model. The MF-DA field is then obtained by applying an iterative smoothing algorithm to the randomly generated field. More specifically, the technique we used visits all the points of the displacement field and updates their values through the iterated conditional mode (ICM) algorithm detailed in [16]. Specifically, when the ICM algorithm starts, all the pixels (x, y) of the displacement field are randomly visited and their displacement vectors updated by trying to minimize the potential function (18). Specifically, a local minimum is sought by letting

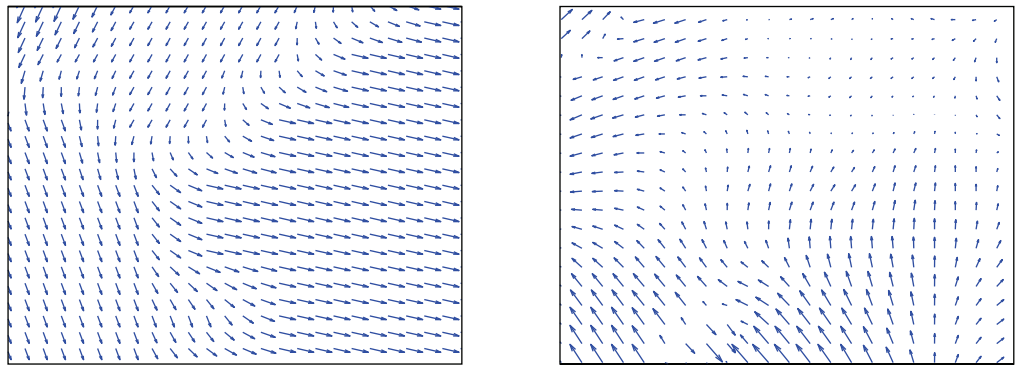
$$\mathbf{f}_{\text{opt}\,(x,y)} = \underset{\mathbf{f} \in (L \times L)}{\arg\min} \sum_{(\tilde{x},\tilde{y}) \in N(x,y)} V_{((x,y),(\tilde{x},\tilde{y}))}. \tag{19}$$

Note that in the above equation, the displacements of the pixels in the neighborhood of (x, y) are fixed, hence resulting in a local minimization of the Gibbs potential. After each pixel is visited and the corresponding displacement gets updated, a new iteration starts. The algorithm ends when no new modification is introduced for a whole iteration, which is usually the case after 7-8 iterations.

As for the LPCD DAs, we considered a multiresolution version of the MF-DA, where the full-resolution version of the the displacement field is built by interpolating the displacement field obtained by applying the MF-DA at a resolution level L. In Figures 5(a) and 5(b), two examples of displacement fields generated with the MF-DA model are shown, using respectively, the parameters $L = 6$, $\sigma = (1, 1)$, $c = 6$ and $L = 4$, $\sigma = (7, 7)$, $c = 18$. With MF-DA, it is possible to obtain larger displacement vectors than with the LPCD attacks (due to the high value of the c parameter), while keeping the distortion invisible, thanks to the ability of the iterative conditional mode to generate a very smooth field, as we can see from Figure 5. A pseudocode description of the MF-DA is provided by Algorithms 3, 4, and 5.

4.2. Perceptual analysis

In order to evaluate the potentiality of the MF-DA class of attacks, the perceptual impact of the distortion they generate must be taken into account. From a perceptual point of view, MRF DAs have a different behavior for different values of L, σ, and c, in particular, the image quality increases if the attacks are generated at a lower level of resolution but, in the meantime, the number of possible distortions decreases.



(a) (b)

Figure 5: Examples of displacement fields generated with MRF DA's: (a) MRF with $L = 6$, $\sigma = 1$, and $c = 6$; (b) MRF with $L = 4$, $\sigma = 7$, and $c = 18$.

1. Read image to be attacked, read level of resolution L, read standard deviation σ, read c
2. dim = size(image)/2^L {size of the low resolution displacement fields δ_h and δ_v}
3. Initialize matrices δ_h and δ_v with random values in the interval $[-c, c]$
4. $\text{diff}_h = \delta_h$
5. $\text{diff}_v = \delta_v$
6. *while* diff_h and diff_v are $\neq 0$ *do*
7. $\quad \text{temp}_h = \delta_h$
8. $\quad \text{temp}_v = \delta_v$
9. $\quad$ row = randperm(dim);
10. $\quad$ col = randperm(dim);
11. $\quad$ *for* $k = 1$: dim *do*
12. $\quad\quad$ *for* $h = 1$: dim *do*
13. $\quad\quad\quad i = \text{col}(1, k)$;
14. $\quad\quad\quad j = \text{row}(1, h)$
15. $\quad\quad\quad [sx, sy] = V_{\text{opt}}(i, j, \delta_h, \delta_v, \sigma, \text{dim})$ {Find the optimum displacements sx and sy, i.e. the ones minimizing the potential function}
16. $\quad\quad\quad \delta_h(i, j) = sx$
17. $\quad\quad\quad \delta_v(i, j) = sy$
18. $\quad\quad$ *end for*
19. $\quad$ *end for*
20. $\quad \text{diff}_h = \delta_h - \text{temp}_h$
21. $\quad \text{diff}_v = \delta_v - \text{temp}_v$
22. *end while*
23. Resize the displacement fields given by δ_h and δ_v to the image size through bicubic interpolation provided by the matlab function *imresize* {to obtain the high resolution displacement fields Δ_h an Δ_v}
24. *for* $i = 1$: size(image) *do*
25. $\quad$ *for* $j = 1$: size(image) *do*
26. $\quad\quad Z(i, j) = Y(i + \Delta_h(i, j), j + \Delta_v(i, j))$ Apply the displacement fields to the image, to obtain the attacked image Z, by means of bicubic interpolat}
27. $\quad$ *end for*
28. *end for*

Algorithm 3: MF-DA-based model.

After a visual inspection conducted on a set of images, we found, for each level of resolution, the maximum value of the σ components and c that can be used while keeping the distortion invisible. Specifically, we found that, in case of images of size 512×512, the larger perceptually admissible displacements are obtained by using $L = 6$, $\sigma = 1$, $c = 6$, $L = 5$, $\sigma = 3$, $c = 8$, and $L = 4$, $\sigma = 7$, $c = 18$, ($\sigma = \sigma_x = \sigma_y$).

In Figure 6, two examples of images distorted with an MF-DA attack applied at different levels of resolution are

(a) (b) (c) (d)

Figure 6: Example of two images attacked with the MF-DA model: (a) original image; (b) attacked image with $L = 6$; (c) original image; (d) attacked image with $L = 4$.

1. Read position of the pixel (i, j), matrices of displacement fields δ_h and δ_v, standard deviation σ
2. $sx_{\text{temp}} = \delta_h(i, j)$
3. $sy_{\text{temp}} = \delta_v(i, j)$
4. $V_{\text{init}} = \text{Gibbs}(i, j, sx, sy, \delta_h, \delta_v)$ {Initial potential}
5. *for* $sx = -i + 1 : \dim - i$ *do*
6. *for* $sy = -j + 1 : \dim - j$ *do*
7. $V_{\text{temp}} = \text{Gibbs}(i, j, sx, sy, \delta_h, \delta_v)$
8. *if* $V_{\text{temp}} < V_{\text{init}}$ *then*
9. $V_{\text{init}} = V_{\text{temp}}$
10. $sx_{\text{temp}} = sx$
11. $sy_{\text{temp}} = sy$
12. *end if*
13. *end for*
14. *end for*
15. $sx = sx_{\text{temp}}$
16. $sy = sy_{\text{temp}}$
17. *return* sx and sy

Algorithm 4: Function $V_{\text{opt}}(i, j, \delta_h, \delta_v, \sigma, \dim)$.

1. Read position of the pixel (i, j), displacements sx and sy, matrices of displacement fields δ_h and δ_v
2. $N(i, j) = [(i - 1, j); (i + 1, j); (i, j - 1); (i, j + 1)]$ $N(i, j)$ is a first order neighborhood system associated with the pixel (i, j)
3. $$V_{((i,j),(\tilde{i},\tilde{j}))} = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left\{ -\left[\frac{(sx - \delta_h(\tilde{i},\tilde{j}))^2}{2\sigma_x^2} + \frac{(sy - \delta_v(\tilde{i},\tilde{j}))^2}{2\sigma_y^2} \right] \right\}$$
4. $$\text{Potential} = \sum_{(\tilde{i},\tilde{j}) \in N(i,j)} V_{((i,j),(\tilde{i},\tilde{j}))}$$
5. *return* Potential

Algorithm 5: Potential funtion $\text{Gibbs}(i, j, sx, sy, \delta_h, \delta_v)$.

shown: in the Barbara image, the MRF is applied at a lower level of resolution ($L = 6$), while in the Lena image, the distortion is generated at a higher level of resolution ($L = 4$). In both cases by comparing the original image (on the left) with the attacked one (on the right), we can notice a slightly perceptible distortion that is, however, not annoying due to the smoothness constraints of the field (the distortion is not visible if only the attacked image is provided so that the comparison with the original image is not possible).

Regarding the cardinality evaluation of this new class of DAs, in principle all the displacement fields are allowed, with the most annoying distortions corresponding to very low probabilities (and thus very large Gibbs potential). In order to evaluate the cardinality of the MF-DA class, then,

Table 2: Value of the parameters used for the experiments.

	Parameter	Value
Stirmark	b	0
	d	0
	i	0
	o	0
	R	0.1
MF-DA	c	$\frac{\dim}{2}$
DCT system	k	5
	L	25000
	M	16000
DWT system	k	2

a first step would be to calculate the entropy rate of the field. However, this is a prohibitive task given that no technique is known to calculate the entropy rate of even the simplest MRFs.

5. DESYNCHRONIZATION PROPERTIES OF THE VARIOUS DAS

In this section, we evaluate the desynchronization capability of the various classes of attacks. To do so, two very simple watermarking algorithms were implemented and the ability of the various DAs to inhibit watermark detection was evaluated. The source image database used for the experiments includes the six standard images: Baboon, Barbara, Boats, Goldhill, Lena, and Peppers. The source image database and the software we used for the experiments are available on http://www.dii.unisi.it/~vipp.

The tested algorithms include

(i) blind additive spread spectrum in the frequency domain (BSS-F),

(ii) blind additive spread spectrum in the wavelet domain (BSS-W).

In both the systems, the watermark consists of a sequence of n_b bits $X = \{x(1), x(2), \ldots, x(n_b)\}$; each value $x(i)$ being a random scalar that is either 0 or 1 with equal probability.

In the BSS-F algorithm, the watermark is inserted into the middle frequency coefficients of the full-frame DCT domain. The DCT of the original image is computed, the frequency coefficients are reordered in a zig-zag scan, and the first $L + M$ coefficients are selected to generate a vector $W = \{t(1), t(2), \ldots, t(L), t(L+1), \ldots, t(L+M)\}$. Then, in order to obtain a tradeoff between perceptual invisibility and robustness to image processing techniques, the lowest L coefficients are skipped and the watermark X is embedded in the last M coefficients $T = \{t(L+1), \ldots, t(L+M)\}$ to obtain a new vector $T' = \{t'(L+1), \ldots, t'(L+M)\}$ according to the following rule:

$$\begin{aligned} T' &= T + k\text{PN} \quad \text{if bit} = 0, \\ T' &= T - k\text{PN} \quad \text{if bit} = 1, \end{aligned} \tag{20}$$

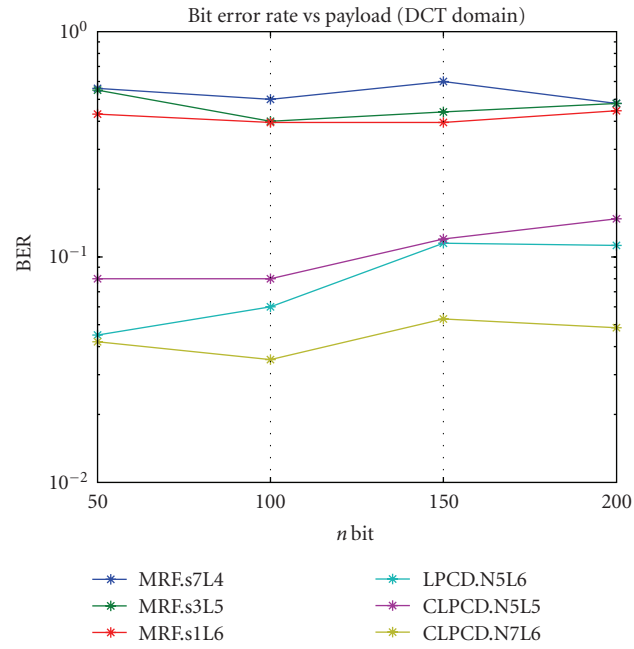


Figure 7: Desynchronization capabilities of the various DAs against the DCT domain system.

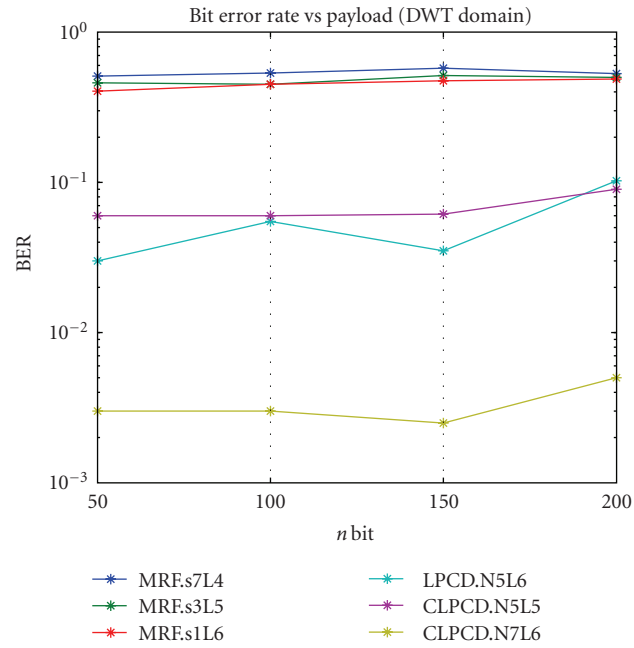


Figure 8: Desynchronization capabilities of the various DAs against the DWT domain system.

where k is the embedding strength and PN is a uniformly distributed pseudorandom sequence of 1 and -1. (20) refers to the embedding of one bit, the extension to multiple bits consists of applying (20) for each bit considering each time a different subset of 0 T and a different PN sequence (a more-detailed description of the watermark embedding is given by the Algorithm 4).

In watermark detection, the DCT is applied to the watermarked (and possibly attacked) image, the DCT coefficients

1. Read image to be watermarked, length of the watermark n_b, energy of the watermark k, seed key, L, M
2. Generate a random n_b long message
3. Perform full-frame DCT
4. Reorder the DCT coefficients into a zig-zag scan
5. Select the coefficients: $T_L^{L+M} = \{t(L), t(L+1), \ldots, t(L+M)\}$ middle frequency coefficients to be watermarked}
6. *for* bit = 1: n_b *do*
7. Generate an antipodal PN sequence of length $l$bit $= M/n_b$
8. $a = (\text{bit} - 1) * l\text{bit} + 1$ and $b = (\text{bit} - 1) * l\text{bit} + l\text{bit}$
9. *if* bit = 0 *then*
10. $\hat{T}_a^b = T_a^b + k\text{PN}$
11. *else*
12. $\hat{T}_a^b = T_a^b - k\text{PN}$
13. *end if*
14. *end for*
15. Reinsert the vector $\hat{T}$ in the zig-zag scan
16. Perform inverse scan
17. Perform inverse full frame DCT
18. Save watermarked image and message

Algorithm 6: DCT domain watermarking: embedding.

1. Read watermarked image, seed key, length of the watermark n_b and load inserted message {needed to evaluate bit error rat}
2. Perform full frame DCT transform
3. Reorder the DCT coefficients into a zig-zag scan
4. Select the coefficients: $T_L^{*L+M} = \{t(L), t(L+1), \ldots, t(L+M)\}$ {middle frequency watermarked coefficients}
5. *for* bit = 1: n_b *do*
6. Generate an antipodal PN sequence of length $l$bit $= M/n_b$
7. Compute the correlation coefficient as expressed in (21) between PN and T_a^{*b} where $a = (\text{bit} - 1) * l\text{bit} + 1$ and $b = (\text{bit} - 1) * l\text{bit} + l\text{bit}$
8. *end for*
9. *for* bit = 1: n_b *do*
10. *if* correlation(bit) > 0 *Then*
11. extracted_message(bit) = 0
12. *else*
13. extracted_message(bit) = 1
14. *end if*
15. *end for*
16. *return* Bit Error Rate

Algorithm 7: DCT domain watermarking: decoding.

are reordered into a zig-zag scan, and the coefficients from the $(L+1)$th to the $(L+M)$th are selected to generate a vector $T^* = \{t(L+1), \ldots, t(L+M)\}$. For each bit, the correlation coefficient between the corresponding subset of the T^* vector and a new PN sequence is evaluated and compared to a threshold (equal to 0) to recover the embedded bit.

The correlation coefficient is evaluated in the following way:

$$r(A,B) = \frac{\sum_{i=1}^{n} (A(i) - \mu(A))(B(i) - \mu(A))}{\sqrt{\left(\sum_{i=1}^{n} (A(i) - \mu(A))^2\right)\left(\sum_{i=1}^{n} (B(i) - \mu(B))^2\right)}}, \tag{21}$$

where A and B are two vectors of same size n and μ is the mean operator. The decision rule states that,

$$\begin{aligned} \text{bit} &= 0 \quad \text{if } r > 0, \\ \text{bit} &= 1 \quad \text{if } r < 0. \end{aligned} \tag{22}$$

In the BSS-W watermarking system, the watermark is added to the DWT coefficients of the three largest detail (i.e., LH, HL, HH) subbands of the image. The embedding and decoding functions are implemented in the same way of the previous system but the watermark is inserted in the wavelet coefficients obtained with a one-step wavelet decomposition. A more-detailed description of the two watermarking systems is given by the Algorithms 6, 7, 8, and 9.

1. Read image to be watermarked, length of the watermark n_b, energy of the watermark k, seed key
2. Generate a random n_b long message
3. Perform a one step wavelet decomposition using Haar filter
4. Reorder the LH, HL and HH components into a vector T
5. *for* bit = 1: n_b *do*
6. Generate an antipodal PN sequence of length $l$bit = size$(T)/n_b$
7. $a = (\text{bit} - 1) * l\text{bit} + 1$ and $b = (\text{bit} - 1) * l\text{bit} + l\text{bit}$
8. *if* bit = 0 *then*
9. $\hat{T}_a^b = T_a^b + k\text{PN}$
10. *else*
11. $\hat{T}_a^b = T_a^b - k\text{PN}$
12. *end if*
13. *end for*
14. Perform a one step inverse wavelet decomposition using Haar filter
15. Save watermarked image and message

Algorithm 8: DWT domain watermarking: embedding.

1. Read watermarked image, seed key, length of the watermark n_b and load inserted message {needed to evaluate bit error rate}
2. Perform a one step wavelet decomposition using Haar filter
3. Reorder the LH, HL and HH components into a vector T^*
4. *for* bit = 1: n_b *do*
5. Generate an antipodal PN sequence of length $l$bit = size$(T)/n_b$
6. Compute the correlation coefficient as expressed in (21) between PN and T_a^{*b} where $a = (\text{bit} - 1) * l\text{bit} + 1$ and $b = (\text{bit} - 1) * l\text{bit} + l\text{bit}$
7. *end for*
8. *for* bit = 1: n_b *do*
9. *if* correlation(bit) > 0 *then*
10. extracted_message(bit) = 0
11. *else*
12. extracted_message(bit) = 1
13. *end if*
14. *end for*
15. *return* Bit Error Rate

Algorithm 9: DWT domain watermarking: decoding.

The six standard images were watermarked with the systems described above with different payloads and then attacked with RBA and the two new classes of attacks. Each image is attacked with a different realization of the field. In Table 2, the values of the parameters used for the experiments are shown. Figures 7 and 8 show the ability of the RBA and of the two new DAs to inhibit correct decoding. The average of the bit-error rate obtained for the six images is plotted versus different values of the payload for both the watermarking systems.

For both the systems, the RBA attack is not able to prevent a correct watermark decoding, in fact, the RBA plot is not visible in the figures because the bit-error rate is always equal to zero. A more powerful class of DAs is the LPCD DAs that in both the systems gives a bit-error rate much higher than the RBA attack. The MF-DA always results in a very high bit-error rate also applying the attack to a lower level of resolution.

6. CONCLUSION

In this paper, we introduced two new classes of desynchronization attacks that extend the class of local geometric attacks so to allow for more powerful attacks with respect to classical RBA. The effectiveness of the new classes of DAs is evaluated from different perspectives including perceptual intrusiveness and desynchronization efficacy. The experimental results showed that the two new classes of attacks are more powerful than the local geometric attacks proposed so far.

This work can be seen as a first step towards the characterization of the whole class of perceptually admissible DAs, which in turn is an essential step towards the development of a new class of watermarking systems that can effectively cope with them.

Future works may include the development of a perceptual metric suited for geometric distortions and the use of new potential functions.

ACKNOWLEDGMENT

This work was supported by the Italian Ministry for University and Research, under FIRB Project no. RBIN04AC9W: "Image watermarking in the presence of geometric attacks, theoretical analysis, and development of practical algorithm."

REFERENCES

[1] F. A. P. Petitcolas, "Stirmark benchmark 4.0.," http://www.petitcolas.net/fabien/watermarking/stirmark/.

[2] J. Lichtenauer, I. Setyawan, T. Kalker, and R. Lagendijk, "Exhaustive geometrical search and the false positive watermark detection probability," in *Security and Watermarking of Multimedia Contents V*, vol. 5020 of *Proceedings of SPIE*, pp. 203–214, Santa Clara, Calif, USA, January 2003.

[3] M. Barni, "Effectiveness of exhaustive search and template matching against watermark desynchronization," *IEEE Signal Processing Letters*, vol. 12, no. 2, pp. 158–161, 2005.

[4] S. Voloshynovskiy, F. Deguillaume, and T. Pun, "Multibit digital watermarking robust against local non linear geometrical distortions," in *Proceedings of IEEE International Conference on Image Processing (ICIP '01)*, vol. 3, pp. 999–1002, Thessaloniki, Greece, October 2001.

[5] S. Pereira and T. Pun, "Fast robust template matching for affine resistant image watermarking," in *Proceedings of the 3rd International Workshop on Information Hiding (IH '99)*, vol. 1768 of *Lecture Notes in Computer Science*, pp. 199–210, Dresden, Germany, September-October 1999.

[6] S. Pereira, J. J. K. O. Ruanaidh, F. Deguillaume, G. Csurka, and T. Pun, "Template based recovery of Fourier-based watermarks using log-polar and log-log maps," in *Proceedings of the 6th IEEE International Conference on Multimedia Computing and Systems (ICMCS '99)*, vol. 1, pp. 870–874, Florence, Italy, June 1999.

[7] D. Delannay and B. Macq, "Generalized 2-D cyclic patterns for secret watermark generation," in *Proceedings of IEEE International Conference on Image Processing (ICIP '00)*, vol. 2, pp. 72–79, Vancouver, BC, Canada, September 2000.

[8] M. Kutter, "Watermarking resistance to translation, rotation, and scaling," in *Multimedia Systems and Applications*, vol. 3528 of *Proceedings of SPIE*, pp. 423–431, Boston, Mass, USA, November 1998.

[9] C.-Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, M. L. Miller, and Y. M. Lui, "Rotation, scale, and translation resilient watermaking for images," *IEEE Transactions on Image Processing*, vol. 10, no. 5, pp. 767–782, 2001.

[10] F. A. P. Petitcolas and R. J. Anderson, "Evaluation of copyright marking systems," in *Proceedings of the 6th IEEE International Conference on Multimedia Computing and Systems (ICMCS '99)*, vol. 1, pp. 574–579, Florence, Italy, June 1999.

[11] N. Merhav, "An information-theoretic view of watermark embedding-detection and geometric attacks," in *Proceedings of the 1st Wavila Challenge Workshop (WaCha '05)*, Barcelona, Spain, June 2005.

[12] M. Barni, A. D'Angelo, and N. Merhav, "Expanding the class of watermak de-synchronization attacks," in *Proceedings of the 9th ACM Workshop on Multimedia and Security (MM-Sec '07)*, pp. 195–204, Dallas, Tex, USA, September 2007.

[13] A. D'Angelo, G. Menegaz, and M. Barni, "Perceptual quality evaluation of geometric distortions in images," in *Human Vision and Electronic Imaging XII*, vol. 6492 of *Proceedings of SPIE*, p. 12 pages, San Jose, Calif, USA, January 2007.

[14] T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York, NY, USA, 1991.

[15] S. Li, *Markov Random Field Modeling in Computer Vision*, Springer, London, UK, 1995.

[16] J. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society B*, vol. 48, no. 3, pp. 259–302, 1986.