Still Image Watermarking Robust to Local Geometric Distortions

Jean-Luc Dugelay *Senior Member, IEEE*, Stéphane Roche, Christian Rey and Gwenaël Doërr *Member, IEEE*

***Abstract*— Geometrical distortions are the Achilles heel for many watermarking schemes. Most of countermeasures proposed in the literature only address the problem of global affine transforms (e.g. rotation, scaling and translation). In this paper, we propose an original blind watermarking algorithm robust to local geometrical distortions such as the deformations induced by Stirmark. Our method consists in adding a predefined additional information to the useful message bits at the insertion step. These additional bits are labeled as resynchronization bits or reference bits and they are modulated in the same way as the information bits. During the extraction step, the reference bits are used as anchor points to estimate and compensate for small local and global geometrical distortions. The deformations are approximated using a modified basic optical flow algorithm.**

***Index Terms*— image watermarking, geometrical distortions, attacks, stirmark, counterattacks, robustness, resynchronization, optical flow, self-similarities.**

I. Introduction

IMAGE watermarking is now a major field. Basically, digital watermarking [1], [2] allows owners or providers to hide an invisible and robust message inside multimedia content, often for security purposes, in particular owner or content authentication. There is a complex trade-off between three parameters in digital watermarking: data payload, fidelity and robustness. Data payload can be defined as the number of useful bits that can be hidden in an image. It should be noted that, most of the time, data payload depends on the size of the host image. The more host samples available, the more bits that can be hidden. The capacity is consequently often given in terms of ratio (i.e. bits per sample). Fidelity can be seen as the perceptual similarity between the original and watermarked image. The modifications introduced by the watermarking process should be imperceptible. And finally, robustness means that the retriever is still able to recover the hidden message even if the watermarked content has been altered after embedding. Until now, most of the proposed watermarking algorithms, derived from classical schemes such as [3]–[6], are mainly robust against usual image processing such as low pass filtering or JPEG compression. However most of them are still weak against geometrical distortions and malicious attacks [7], [8] (e.g. *Stirmark* [9], [10], collusion attacks, *Dewatermarking Attack* [11], etc.). It is quite easy to see that data payload, fidelity and robustness are often in conflict. One may want to increase the watermarking strength in order to increase the robustness, but this also results in a more perceptible watermark. On the other hand, one can increase the data payload by decreasing the redundancy of each hidden bit, but this is counterbalanced by a loss of robustness. As a result, a trade-off has to be found and it is often tied to the targeted application.

In this paper, we present an original approach to compensate for local and global geometrical distortions, in particular the random deformations generated by the *Stirmark* attack. The proposed method operates directly in the spatial domain, and does not require for the extraction step any a priori knowledge neither on the original image, nor about the hidden message (i.e. blind extraction). The paper is organized as follows: in Section II, we first introduce the problem of watermarking desynchronization and shortly give an overview of previous countermeasures proposed in the literature; in Section III, we describe the baseline of our watermarking scheme based on self-similarities; in Section IV, we present the resynchronization method; Section V reports the performances of the proposed approach, but also its limitations against large geometrical attacks; a complementary method is consequently briefly described in Section VI; and finally, we conclude and draw perspectives in Section VII.

II. Problem Context

A. Effect of Geometric Distortions

A watermarking scheme should not only be robust to photometric transformations, but also to geometrical distortions. We distinguish between global and local geometrical manipulations because the resynchronization methods significantly differ. In the first case, there is enough number of samples to estimate the parameters of the transformation, whereas in the second case, the number of samples is limited since the transformation must be estimated locally. The most commonly used global geometrical transformations are rotation of a few degrees, translation, scaling and cropping. These manipulations are very popular in image processing and do not necessarily lead to visual inconvenience. As for the local transformations, they generally depend, either on the position of the pixel in the image, or on a pseudo-random process (i.e. *Stirmark*). The continuity of the borders of the areas undergoing the transformations is ensured by a bilinear interpolation function.

Manuscript received ???; revised ???. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Gaurav Sharma.

The authors have all been with the Eurécom Institute, Sophia-Antipolis, France at some time. Prof. Dugelay can be reached by email at dugelay@eurecom.fr.

Digital Object Identifier 10.1109/TIP.0000.000000

From the image watermarking point of view, geometrical distortions mainly introduce desynchronization between the encoder (i.e. the watermark hidden into the image) and the decoder. The watermark signal is still present, but the decoder is no longer able to detect or extract it. The explanation is that the watermark usually undergoes the same (or dual) distortions as the image. As a result without reference mark or the original image, the detector tries to extract the watermark from its initial locations which are different from current ones.

B. State of the Art of Existing Resynchronization Methods

In this section, we present the outline of classical resynchronization methods. The difficulties and the techniques to detect or extract a watermark from a distorted image depend on the mode of extraction and obviously the nature of the geometrical deformations. For example, in a non-blind mode, the original image can be used to identify the geometrical deformations. In this case, the problem amounts to calculating an optical flow between the attacked image and the original one. It is then enough to apply the pseudo-reverse transformations to compensate for the distortions. For blind watermarking schemes the problem is radically different: the watermark and the original image are unknown. Obviously this context is very constraining insofar as we do not know which manipulations the image underwent.

1) Exhaustive search: The simplest method for watermark detection after geometrical distortions is an exhaustive search [12]. This approach consists simply in inverting each hypothetical geometrical deformation that might have been applied to the watermarked image, and then applying the watermark detector once for each possible distortion parameter. Obviously, this method is feasible for a restricted number of hypothetical deformations (e.g. scaling, rotation, translation), but it rapidly becomes intractable as the number of possible distortions increases. Furthermore, it tends to largely increase the false positive probability [13].

2) Methods resorting to the original image: Dong et al. [14] as well as Davoine et al. [15] propose a method resorting to the original image to compensate for the geometrical deformations induced by *Stirmark*. It consists of a preprocessing step directly performed on the attacked image, independent of the watermarking algorithm. A regular triangular tessellation is applied on both original and watermarked images. The compensation for geometrical distortions is performed by slightly shifting different vertices of the attacked image tessellation to minimize the quadratic error between each original triangle and the corresponding attacked triangle (Figure 1). Once the image has been compensated, the watermark can be detected or extracted. This method is very efficient only in the case of minor deformations. If the image undergoes major local or global distortions, the compensation can provide for some mismatching between meshes and the retriever is unable to extract the watermark. Moreover, the need for the original image (or some kind of information pre-extracted from it) restricts notably the interest of this approach. In order to reduce the amount of a priori information necessary for the

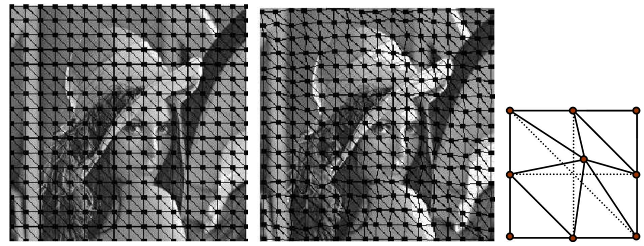


Fig. 1. Compensation for geometrical distortions using the original image. Left image: original image with triangular tessellation. Right image: distorded image after geometrical compensation.

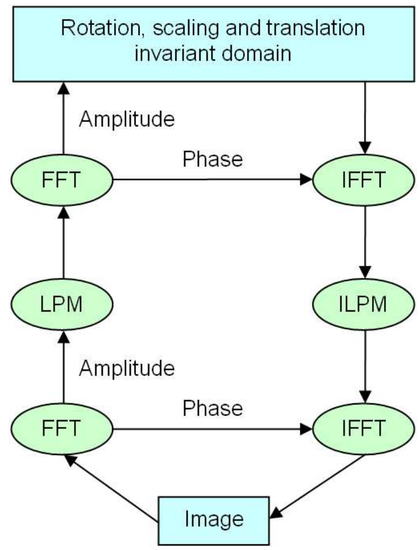
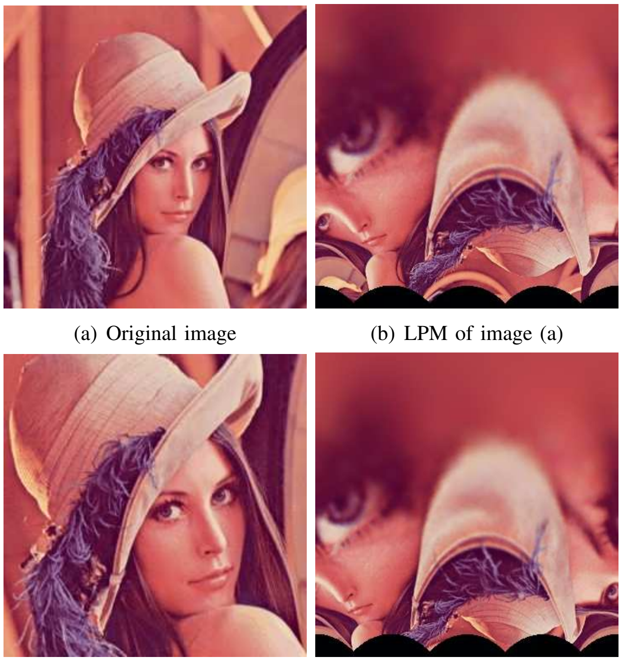


Fig. 2. Principle of building a rotation, scale and translation invariant subspace from a digital image, using the Fourier-Mellin transform.

extraction step, Johnson et al. [16] propose inverting affine transformations using only a set of feature points extracted from the original image. So the resynchronization process consists in estimating the best affine transformation (in the least mean square sense) by using the correspondences between feature points extracted from the original image and those from the modified one.

3) Embedding the watermark in an invariant subspace: Another solution consists in embedding the watermark in a geometrical invariant subspace. Coltuc and Bolon [17] suggest using histogram specification to hide a watermark invariant to geometrical distortions. Ruanaidh and Pun [18], and Lin et al. [19] propose a watermarking scheme based on the Fourier-Mellin transform [20] (c.f. Figure 2). A rotation, scale and translation invariant domain is obtained using the log-polar mapping (LPM) and the Fourier transform invariance properties to translations. In the resulting log-polar map, rotations and scalings come down to translations (c.f. Figure 3).

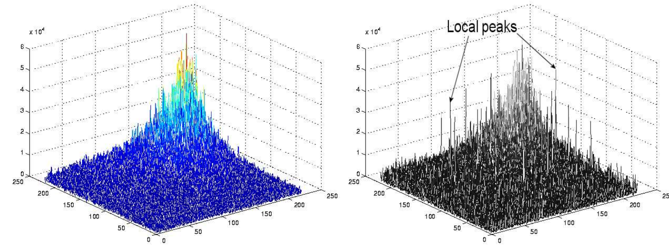
In practice, this solution can be implemented for simple affine transformations, but it is inapplicable as soon as the image undergoes local geometrical deformations. Moreover problems of approximation due to the discrete nature of the images, plus the reduction of the embedding space make the watermark weakly resistant to low-pass filtering and lossy



(a) Original image (b) LPM of image (a)

(c) Manipulated image (d) LPM of image (c)

Fig. 3. Examples of LPM transformation in the spatial domain. The log-polar mapping converts rotation and scaling to translation.



(a) FFT of original image (b) FFT of watermarked image

Fig. 4. Example of resynchronization method resorting to a template (image watermarked by *Digimarc* [32]).

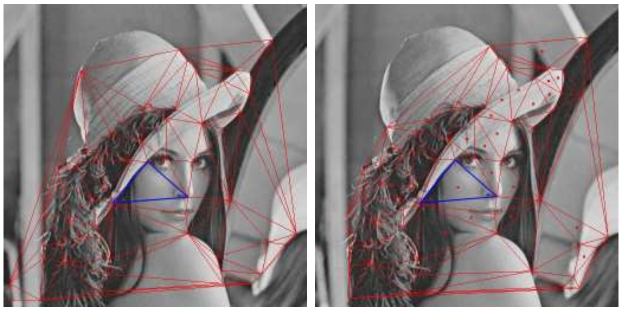
compression.

Alghoniemy and Tewfik [21] present another approach where the watermarking space is defined as a canonical, normalized space based on the geometric image moments. These moments are used to transform the image into a form that is independent to rotation, scale, and horizontal/vertical reflection. The watermark is embedded in this space, and then the inverse transformation is applied to obtain the final watermarked image. During detection, the moments are calculated again and used to estimate the normalization parameters. Once the image is normalized, the watermark can be detected.

Rather than embedding the watermark in an invariant subspace, Solachidis and Pitas [22] propose creating a self-similar watermark, and to embed it then into the DFT domain. Thus, their method is robust to translation, since they do not affect the DFT magnitude. Since the watermark is made up of S identical sectors, the detection is possible even after a $\frac{2k\pi}{S}$ degree rotation. The self-similar properties of the watermark also allows for the reduction of the number of different frequency sampling steps where the detection should be performed when the image has been cropped and scaled.

4) Methods using a template or a periodic watermark: Currently, one of the most common resynchronization techniques consists in inserting an additional watermark or template (also called a registration pattern [23], [24]) into the image. Basically, this template is used as a reference to detect and compensate for geometrical deformations such as affine transforms. Gruhl and Bender [25] were among the first authors to suggest such a scheme. Their idea was to embed multiple cross shapes into the least significant bits (LSB) of the image. Thus, any geometrical transformation applied to the image will be reflected in the shape and position of the embedded crosses. This information could be used to determine the transformation. The major drawback of this method was its weak robustness against photometric transformations. Pereira and Pun [26] suggest embedding a pseudo-random template in a ring corresponding to the middle frequencies of the image spectrum. The template is generated by increasing the magnitude of selected coefficients and creating local peaks. The synchronization is performed by estimating the affine transformation which allows for the matching of the initial template with the detected location of local maxima. Kutter [27] as well as Deguillaume et al. [28] and Voloshynovskiy et al. [29] propose a similar methods which inserts a periodic pattern in the spatial domain. One major advantage of these methods is their effectiveness to address synchronization of affine transformations, but this kind of pattern is unfortunately, for both watermarker and attacker, easily detectable in frequency domain (c.f. Figure 4). It is then relatively easy for a malicious party to remove these peaks (in particular using the *Template Removal Attack* [30]), thus depriving the detector of any means of resynchronization. Therefore, the template itself must provide sufficient security against erasure and attacks to be really useful. More recently, Delannay and Macq [31] proposed an informed coding approach against this kind of attack. Their technique consists in modulating the resynchronization pattern with an image-dependent secret binary mask in order to prevent erasure of the specific peaks in DFT. The mask is obtained from a signal-dependent partition which is robust against rotation and scaling.

5) Methods using feature points of the image: The last class of resynchronization techniques (also called second generation schemes [33]) uses image content to recover the watermark after geometrical transformations. The method suggested by Bas et al. [34] consists in extracting feature points of the image using a Harris detector [35] and performing a Delaunay tessellation on the set of points (c.f. Figure 5). For each triangle of the tessellation, an affine transform is performed on the watermark pattern (a right-angled isosceles triangle) to map the triangle in which it will be inserted. The deformed watermark is then embedded using a classical additive scheme. The detection of the watermark is dual to the embedding step. A Delaunay tessellation is performed on the attacked image. Each triangle of the tessellation is warped into a right-angled



(a) Original image tessellation (b) Attacked image tessellation

Fig. 5. Resynchronization method using some feature points of the image.

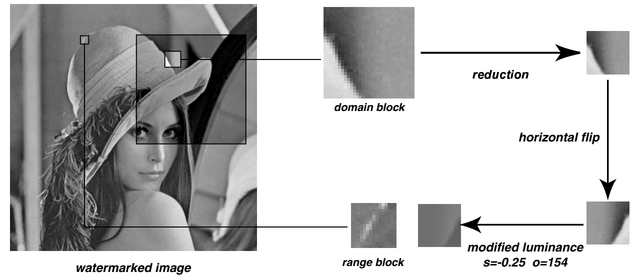


Fig. 6. Self-similarities: an example of matching between range and domain blocks via geometric and photometric affine transformations.

isosceles triangle and a correlation is performed between the original watermark and the different warped triangles. The effectiveness of this approach is mainly based on the stability of the extraction of feature points against various image manipulations, especially on textured images. It is obvious that if a significant number of feature points has been lost, the watermark cannot be detected. Another drawback of this method is that the feature points do not depend on a secret key, so it could be easy for a malicious party to rebuild the same image tessellation and to then perform a collusion attack [36] using the different watermarked triangles to remove the watermark.

Nikolaidis and Pitas [37] propose an original watermarking scheme dedicated for facial color images, using salient points like the eyes and the mouth. The watermark (a pseudo-random binary sequence) is geometrically adapted before embedding, according to the extracted feature positions and the face orientation. The detection step is dual; the watermark is adapted using the features of the tested image and a correlation detector is then employed to decide whether the watermark is present or not.

III. Watermarking Algorithm

In this section, we briefly describe the baseline of our watermarking algorithm [38], [39]. The resynchronization process will be introduced in the next section. The considered approach is derived from fractal image coding theory [40], in particular the notion of self-similarity (c.f. Figure 6). The image is re-expressed as a collection of local similarities up to an affine transform. The main idea is to use invariance properties of fractal coding, such as invariance by affine (geometric and photometric) transformations, to ensure watermark robustness. The extraction mode is blind.

A. Embedding Step

The watermark embedding process can be described by the following three steps: cover generation, formatting and encryption of the message to be hidden, merging of the watermark and the cover.

1) Cover generation: First, a "fractal approximation" I_{approx} is computed from the original image $I_{original}$. The original image is scanned block by block. Those blocks are labeled *range blocks* (block R_i) and have a given dimension $n \times n$ (here 8×8 pixels). Each block R_i is then associated with another block D_i which is similar (modulo a set of possible photometric and geometric transformations) according to a Root Mean Square (RMS) metric defined as follows:

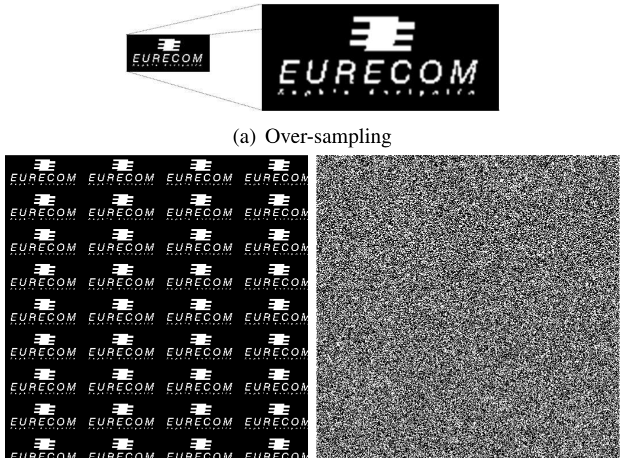
$$RMS(f,g) = \frac{1}{n}\sqrt{\sum_{x=1}^{n}\sum_{y=1}^{n}\Big(f(x,y) - g(x,y)\Big)^2} \quad (1)$$

The block D_i is labeled *domain block* and is searched in a codebook containing Q blocks Q_j. The codebook consits of all possible blocks (up to a precision of 1 pixel) of size 32×32 pixels including inside the search window in the picture itself. Each block Q_j is scaled to match the dimensions of the range block R_i. A set of k geometrically transformed blocks $T_k(Q_j)$ is then built (identity, 4 flips and 3 rotations). For each transformed block $T_k(Q_j)$, a photometric scaling factor s and an offset o are computed to minimize the RMS between the transformed block $g = T_k(Q_j)$ and the range block $f = R_i$.

The range block R_i is substituted by the transformed block $s.T_k(Q_j) + o$ which has the lowest RMS distance. The cover I_{cover} corresponds to the error image that is the signed difference between the original image and its fractal approximation.

$$I_{cover} = I_{original} - I_{approx} \quad (2)$$

2) Formatting of the watermark: The first step consists in converting the message m to be hidden (a string or a logo) into a binary mark. Then, to ensure robustness against small perturbations of the cover, the k bits of the watermark are redundantly distributed. This duplication of the bits consists of two steps. Firstly, the mark is over-sampled by a factor of typically 3 (c.f. Figure 7.a) to produce a low-frequency watermark more resilient to low-pass filtering (e.g., JPEG compression, blur, etc.). Secondly, the over-sampled mark is horizontally and vertically duplicated to create a redundant watermark of the size of the original image (c.f. Figure 7.b). This spatial repetition is necessary to compensate for the possible loss of information due to local image manipulations. Finally, the watermark is encrypted using a XOR between the watermark and a binary pseudo-random sequence generated from a secret key and also over-sampled by a factor of 3 to preserve the low-frequency component of the mark (c.f.



(a) Over-sampling

(b) Message duplicated (c) Message encrypted

Fig. 7. Formatting and encryption of the watermark.

Figure 7.c). The XOR operation allows, on the one hand, to secure the hidden message (e.g. against collusion attack), and on the other, to remove repetitive patterns reducing in this way the psycho-visual impact of the watermark embedding. In order to improve robustness against photometric attacks, an error correction code, in particular block turbo codes [41], can be inserted prior to over-sampling and duplication of the message. Finally we obtain the final binary watermark W.

3) Merging of the watermark and the cover: The last step of the embedding process consists in modulating the cover I_{cover} with the watermark W. The modulation consists in zeroing some of the cover pixels p depending on their sign and the corresponding watermark bit to hide (c.f. Equation (3)). In other words, only one bit out of two on average really hosts hidden information.

$$\begin{cases} \text{If } W(p)=1 \text{ and } I_{cover}(p)>0 \text{ then } \hat{I}_{cover}(p)=I_{cover}(p) \\ \text{If } W(p)=0 \text{ and } I_{cover}(p)<0 \text{ then } \hat{I}_{cover}(p)=I_{cover}(p) \\ \text{Else } \hat{I}_{cover}(p)=0 \end{cases} \quad (3)$$

For visibility reasons, only the low value pixels in I_{cover} will host mark information:

$$\text{If } \left|I_{cover}(p)\right| > \delta_{high} \text{ then } \hat{I}_{cover}(p)=I_{cover}(p) \quad (4)$$

Finally, the modulated cover $\hat{I}_{cover}$ is added to the fractal approximation I_{approx} to produce the watermarked image $I_{watermarked}$. By default, we set the watermarking distortion to about 38 dB.

B. Extraction Step

1) Cover extraction: This step is similar to the cover generation step of the watermark embedding process. An IFS code is calculated from the watermarked image (possibly modified) to obtain $\tilde{I}_{approx}$ and extract the cover of the watermark $\tilde{I}_{cover}$. Then, the obtained cover is thresholded in order to eliminate low (near zero) and high (greater than δ_{high}) values which carry no information related to the watermark. Finally this

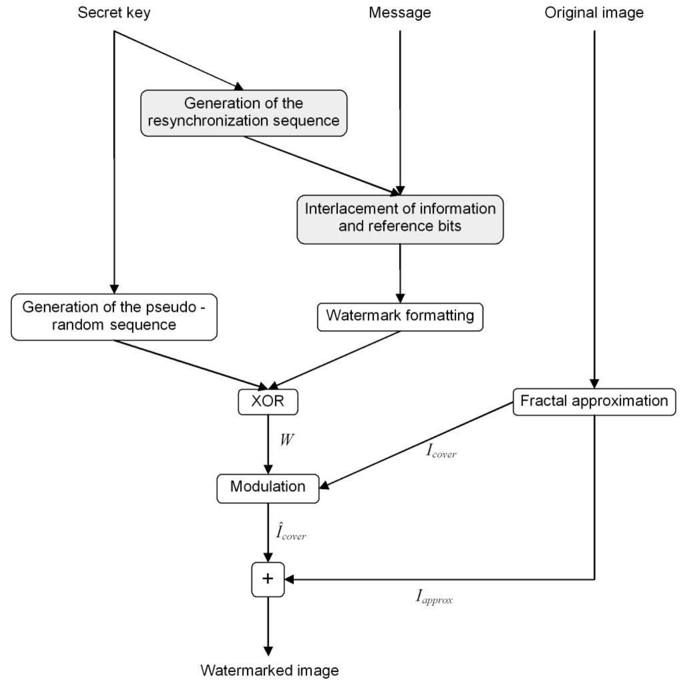


Fig. 8. Flow chart of the embedding process of the proposed watermarking scheme.

cover is decoded according to the modulation rules (e.g. the sign of the pixel cover).

$$\begin{cases} \text{If } \delta_{low} < \tilde{I}_{cover}(p) < \delta_{high} \text{ then } \tilde{W}(p)=1 \\ \text{If } -\delta_{high} < \tilde{I}_{cover}(p) < -\delta_{low} \text{ then } \tilde{W}(p)=-1 \\ \text{Else } \tilde{W}(p)=0 \end{cases} \quad (5)$$

2) Reconstruction of the message: The extracted watermark is then decrypted by reapplying a XOR operation between the watermark and the same binary pseudo-random sequence as performed during the insertion step. Finally, the message is obtained back by exploiting the redundancy: for each k bit of the message, we compute the sum b_k of its values at R_k positions in $\tilde{W}$ corresponding to over-sampling and repetition:

$$\forall k \in [1 \dots K] \qquad b_k = \sum_{r=1}^{R_k} \tilde{W}(p_{k,r}) \quad (6)$$

where $p_{k,r}$ is the set of pixels in $\tilde{W}$ hosting the k^{th} bit of the message. The final value of a bit m_k of the hidden message is determined by "majority vote":

$$\begin{cases} \text{If } b_k < 0 \text{ then } m_k = 0 \\ \text{If } b_k \geq 0 \text{ then } m_k = 1 \end{cases} \quad (7)$$

IV. RESYNCHRONIZATION METHOD

In this section, we introduce our resynchronization module and present how it takes place in the framework previously described in Section III (c.f. Figures 8 and 11).

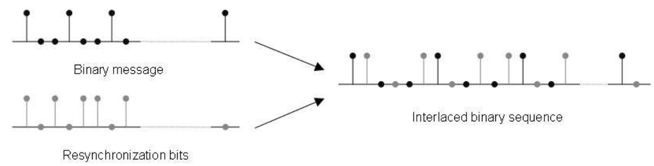


Fig. 9. Interlacing of information and resynchronization bits.

A. Overview of the Method

The proposed method operates directly in the spatial domain and does not require any a priori knowledge of the original image or the hidden message for the extraction step. First of all, our resynchronization technique consists in adding predefined additional information to the useful message bits at the insertion step. These additional bits are labelled as resynchronization bits or reference bits and they are modulated in the same way as the information bits. During the extraction step, the resynchronization bits are used as reference points to estimate and compensate for small local or global geometrical deformations. The distortions are approximated using classical optical flow method.

B. Formatting of the Watermark

The sequence of resynchronization bits is randomly generated using the secret key. Then, the resynchronization bits are uniformly interlaced with the information bits to produce a new binary sequence (c.f. Figure 9). This mono-dimensional sequence is then shaped according to a zigzag path to produce a two-dimensional pattern. The rest of the watermark formatting step is unchanged compared with the method described in paragraph III-A.2; that is to say, the pattern is over-sampled by a factor of 3, then horizontally and vertically duplicated, and finally encrypted.

A trade-off must be found between the number of information (i.e. useful bits) and resynchronization bits. Indeed, if the density of the resynchronization bits was high, the geometrical deformations could be estimated precisely, but the redundancy of information bits would be lower and the message would be less robust (c.f. Figure 10). On the contrary, if the density of the resynchronization bits was low, it would be very difficult to identify and compensate for the geometrical distortions. In this case, the resynchronization process would fail and the message could not be extracted. According to our experience, we suggest to use the same proportion of information and resynchronization bits. In other words, when a user want to embed let say 64 bits, the watermarking system actually insert 128 bits (64 information bits and 64 resynchronization bits).

C. Resynchronization Process at Extraction Step

1) Basic principle: The general idea of the resynchronization method is to map the reference bits with the desynchronized watermark signal before extracting the message. We make the assumption that if we are able to resynchronize the reference bits, the information bits will be resynchronized too. This assumption is realistic insofar as information and resynchronization bits are finely bidimensionally interlaced. Under these conditions, knowing the value and the relative positions of the reference bits, the resynchronization process simply consists in seeking for each portion of the extracted watermark signal (made up of information and reference bits) the best sequence of resynchronization bits (c.f. Figure 12). In practice, most of the non-destructive local geometrical distortions, such as those generated by the *Stirmark* attack, have a limited amplitude and they are almost linear on slight portions of the image. So it is then possible to locally approximate them by simple block translations. Thus the resynchronization process consists in a classical block matching algorithm to calculate the optical flow corresponding to the geometrical attack and to compensate for the deformations of the watermarked image.

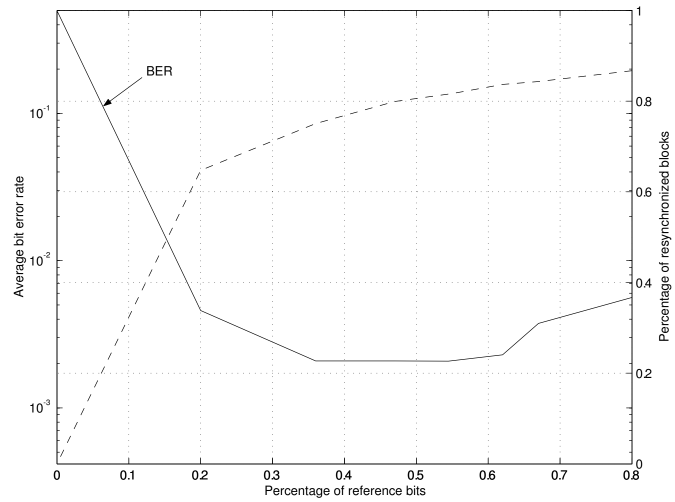


Fig. 10. Resynchronization performance (i.e. percentage of well resynchronized blocks after a translation) *vs.* bit error rate according to the number of reference bits added to the message. The tests have been performed on a database of 75 images.

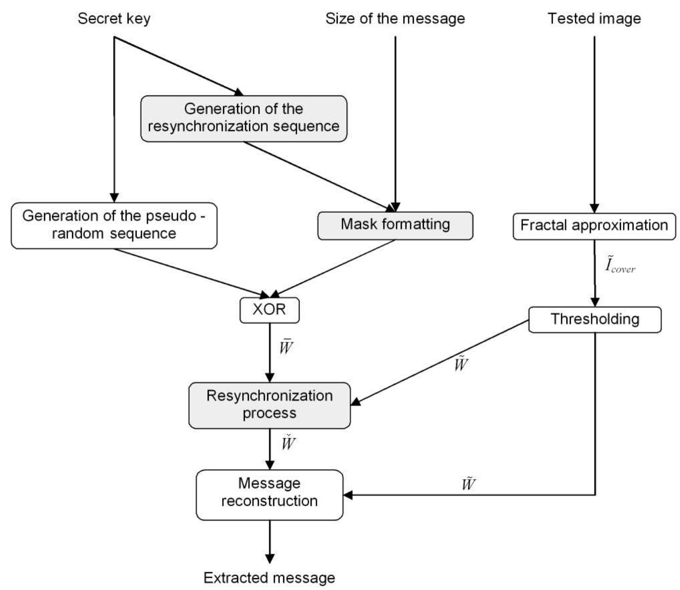


Fig. 11. Flow chart of the extracting process of the proposed watermarking scheme.

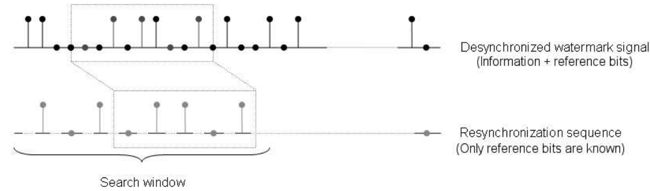


Fig. 12. Basic principle of resynchronization method. One seeks for each portion of watermark signal, the best sequence of resynchronization bits.

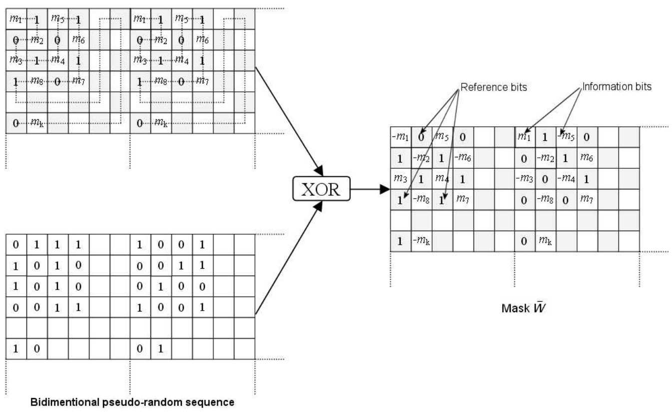


Fig. 13. Example of reference mask generation: let the message $M = (m_1, m_2, m_3, ..., m_k)$ to extract and the resynchronization sequence $R = (1, 0, 1, 0, 1, 1, 0, 1, \ldots, 0)$.

2) Reference mask generation: Once the watermark is modulated with the cover, it is no longer possible to differentiate the information bits from the resynchronization ones. In order to use the resynchronization bits as reference pattern it is necessary to know their value and the way the watermark was shaped. Thus, we create a mask $\bar{W}$ which is used as a reference map indicating precisely the original layout of the information and resynchronization bits, as well as the way in which the pseudo-random sequence was applied. The mask generation is similar to the one of the watermark, except that the information bits (i.e. useful bits) are replaced by labels. These labels are signed (according to the XOR operation between the pseudo-random sequence and the mask) in order to enable later to separate the watermark signal from the pseudo-random sequence during the message reconstruction step.

3) Optical flow: Block matching algorithms [42] are well known in video coding where they are used to estimate the motion vectors between two frames. Within the framework of our resynchronization technique, the block-matching is applied between the reference mask $\bar{W}$ and the watermark cover $\tilde{W}$. The watermark cover $\tilde{W}$ corresponds to the extracted cover $\tilde{I}_{cover}$ which was thresholded to eliminate the extreme values, and demodulated to reveal the values of the hidden bits (i.e. '-1', '1' or '0', zero means that the value of the bit is undefined).

The optical flow is calculated as follows. We consider the ternary watermark cover $\tilde{W}$, resulting from the first step of the retriever, and the reference mask $\bar{W}$. The watermark cover $\tilde{W}$ is divided into blocks whose size is $n \times n$ pixels. We define

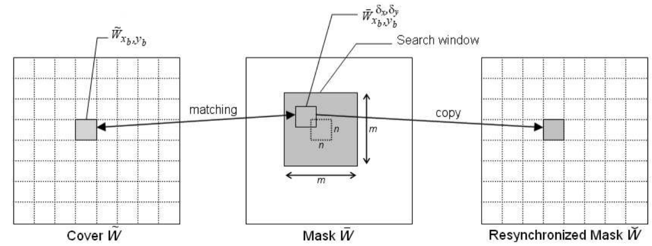


Fig. 14. Block-matching process between the cover $\widetilde{W}$ and the reference mask $\bar{W}$.

a block of W as follows:

$$W^{(\delta_x,\delta_y)}_{x_b,y_b}(x, y) = W(x_b.n + \delta_x + x, y_b.n + \delta_y + y) \qquad (8)$$

where (x_b, y_b) are the indices of the block, (δ_x, δ_y) are the coordinates of the displacement and (x, y) represent the coordinates of an element of the block.

For each block $\tilde{W}_{x_b,y_b}$, a block $\bar{W}^{(\delta_x,\delta_y)}_{x_b,y_b}$ which with the same dimensions is searched within the reference mask $\bar{W}$ by minimizing a cost function (described in the paragraph IV-C.4). An exhaustive search is carried out inside a window of size $m \times m$ pixels, centered on the position of block $\tilde{W}_{x,y}$ (c.f. Figure 14). Thus the dimensions of the search window limit the maximum displacement to $\pm m/2$ pixels around the initial position of the block (i.e. a local distortion of the image greater than $m/2$ pixels could not be compensated). For each candidate block $\bar{W}^{(\delta_x,\delta_y)}_{x_b,y_b}$, we calculate its associated cost $S(\tilde{W}_{x_b,y_b}, \bar{W}^{(\delta_x,\delta_y)}_{x_b,y_b})$. The block $\bar{W}^{(\delta_x,\delta_y)}_{x_b,y_b}$ with the lowest cost is considered as the best candidate and its content is recopied into an array of the same dimensions as the mask, at coordinates of the block $\tilde{W}_{x,y}$. Thus the local distortion is approximated by the translation $\vec{d}(x_b, y_b) = (\delta_{Xbest}, \delta_{Ybest})$.

Once the process has been performed for all the blocks of the watermark cover, one obtains a new reference mask $\check{W}$ adapted to the attacked image. The message is then rebuilt using the cover $\tilde{W}$ and the resynchronized mask $\check{W}$. The decoding principle of repetition codes based on a majority vote is unchanged compared to that described in paragraph III.B.2.

4) Cost function: To determine the best matching criterion between the blocks $\tilde{W}_{x_b,y_b}$ and $\bar{W}^{(\delta_x,\delta_y)}_{x_b,y_b}$, we calculated for each displacement (δ_x, δ_y) in the search window, the number of reference bits correctly resynchronized, the number of reference bits badly paired, the number of undefined pairings (i.e. a reference bit paired with an unknown cover value) and the number of information bits. This experiment shows that only the number of reference bits correctly resynchronized and the number of reference bits badly paired give significant information about the resynchronization. The number of undefined pairings and the number of information bits are approximately constant whatever the tested position is. As a result, they do not influence the block matching process. The cost function consists in calculating the ratio between the number of reference bits badly paired and the reference bits

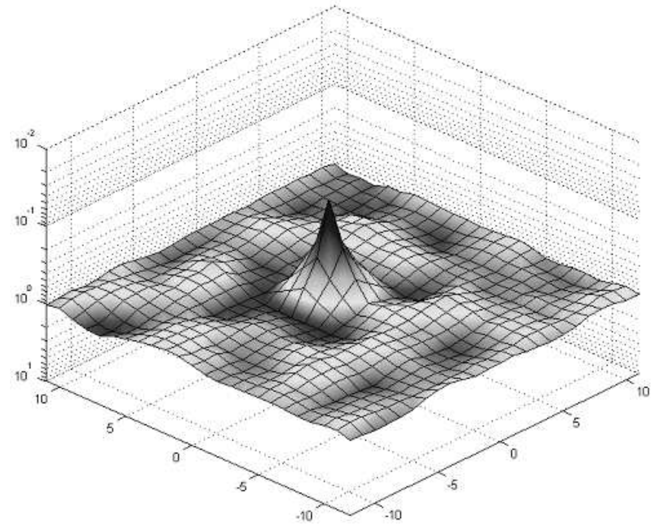


Fig. 15. Cost function according to the tested positions of candidate blocks. In this example, the position of the best candidate block corresponds to the displacement $\vec{d} = (0, 0)$.

correctly resynchronized:

$$S\big(\tilde{W}_{x_b,y_b}, \bar{W}^{(\delta_x,\delta_y)}_{x_b,y_b}\big) = \frac{\displaystyle\sum_{i,j=0}^{(n-1)^2} \Phi\big(\tilde{W}_{x_b,y_b}(i,j), \bar{W}^{(\delta_x,\delta_y)}_{x_b,y_b}(i,j)\big)}{\displaystyle\sum_{i,j=0}^{(n-1)^2} \Psi\big(\tilde{W}_{x_b,y_b}(i,j), \bar{W}^{(\delta_x,\delta_y)}_{x_b,y_b}(i,j)\big)}$$

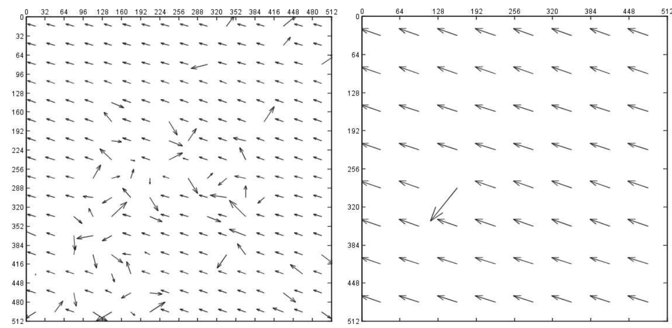
where

$$\Phi\big(A, B\big) = \begin{cases} 1, & \text{if } B \in \{-1, +1\} \text{ and } B = -A \\ 0, & \text{otherwise.} \end{cases}$$

$$\Psi\big(A, B\big) = \begin{cases} 1, & \text{if } B \in \{-1, +1\} \text{ and } B = A \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

5) Improving the estimated displacement field: In this paragraph, we first discuss the influence of the block and search window sizes on the performances of the resynchronization process. Subsequently, we introduce dependencies between neighbor displacements to further enhance the quality of the estimated optical flow.

a) Effectiveness of the resynchronization according to the size of blocks: The size of blocks and the dimension of the search window must be selected judiciously because they determine the effectiveness of the resynchronization process. Blocks which are too large do not allow local geometrical deformations of the image to be compensated, whereas too small blocks do not contain enough reference bits to estimate these deformations correctly. As for the size of the search window, it determines the maximum amplitude of geometrical deformations, as well as the cost in terms of computing time of the resynchronization algorithm. Figure 16 illustrates the influence of the size of blocks during the estimation of the geometrical deformations by the block matching process. In this example, the original image has undergone a slight translation: the left graph represents the optical flow field



(a) Translation (6,2) - blocks 32×32 (b) Translation (6,2) - blocks 64×64

Fig. 16. Effectiveness of the resynchronization according to the size of blocks.

obtained using blocks of 32×32 pixels, and the right one using blocks of 64×64 pixels. Experimentally, the best results were obtained using blocks of 64×64 pixels.

b) Optical flow regulation: The presented resynchronization framework suffers from two major shortcomings:

1) As mentioned previously, the block size has a great influence on the the performances of the block matching based resynchronization process. On one hand, small blocks are likely not to contain enough resynchronization bits to enable a correct registration and thus compensate for small geometric distortions. On the other hand, considering large blocks prevents from estimating finely the geometric distortions. Experimental results have shown that block sizes below 64×64 produce almost random displacement fields.
2) The resynchronization process operates blindly in a best match fashion. There is no constraint between the displacements $\vec{d}(x_b, y_b)$ of neighbor blocks. As a result, the estimated optical flow can contain wrong displacements which significantly interfere with the payload extraction procedure. However, the visual quality of an image distorted by a geometric transformation is determined by its homogeneity [43] i.e. all the displacement fields are not *valid* transformation.

This resynchronization procedure has consequently been slightly modified to further enhance its performances [44]. First, Elastic Graph Matching (EGM) is introduced to smooth the estimated optical flow. Second, a multi-scales approach has been considered to iteratively obtain a denser motion field.

Elastic Graph Matching has been originally introduced in pattern recognition [45]. The idea is to perform a block matching procedure with a smoothness constraint: a rigidity parameter prevents displacements of neighbor blocks from being incoherent. In other words, the goal is to find the set of block displacements $\Delta = \{\vec{d}(x_b, y_b)\}$ which minimizes the global cost function:

$$\mathrm{C}_{\mathrm{total}}(\tilde{W}, \bar{W}, \Delta) = \mathrm{C}_{\mathrm{match}}(\tilde{W}, \bar{W}, \Delta) + \lambda \mathrm{C}_{\mathrm{smooth}}(\Delta) \quad (10)$$

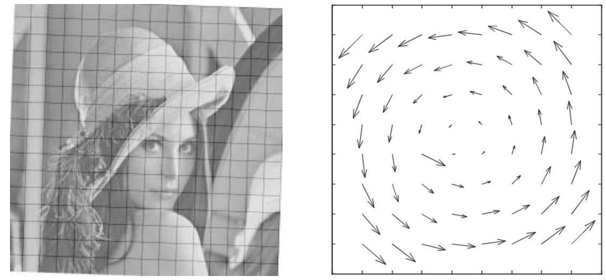
where the parameter λ controls the *rigidity* of the estimated optical flow and $\mathrm{C}_{\mathrm{match}}(.)$ and $\mathrm{C}_{\mathrm{smooth}}(.)$ are two cost functions. The first one is a matching cost function which indicates how well correspond the extracted watermark $\tilde{W}$ and the reference mask $\bar{W}$ considering only the resynchronization bits and assuming that block displacements are given by Δ. It can be thus defined as the sum of the cost functions defined in Equation (9) for the different blocks. It should be noted that for $\lambda = 0$, the resynchronization process comes down to the baseline block matching approach. Therefore, a second term is introduced to ensure that blocks which are not the *best* matching ones are still considered in case they are coherent with the current estimation Δ of the optical flow. To this end, the cost function $\mathrm{C}_{\mathrm{smooth}}(.)$ can be set equal to the sum of the distances between displacements of neighbor blocks. The optical flow is then iteratively updated until a minimum is found.

Nevertheless the presented iterative minimization process can get trapped in a local minimum, especially when small blocks are considered. As a result, a multi-scales approach has been superimposed over the current framework. The basic idea is to start the matching process with large blocks and then to successively consider smaller blocks. Indeed, large blocks (64×64) enable to find a relevant initial estimation of the optical flow since there are many resynchronization bits in each block. Next, the block size is decreased (down to 16×16) to refine the optical flow by permitting more geometric distortions. In practice, the block size is divided twice and the iterative minimization process is launched each time. Furthermore, the estimated optical flow is over-sampled to initialize the procedure for each new scale. The rigidity parameter is also updated accordingly. When the block size decreases, neighbor blocks are nearer and thus neighbor displacements should be more similar i.e. the rigidity parameter should be larger.

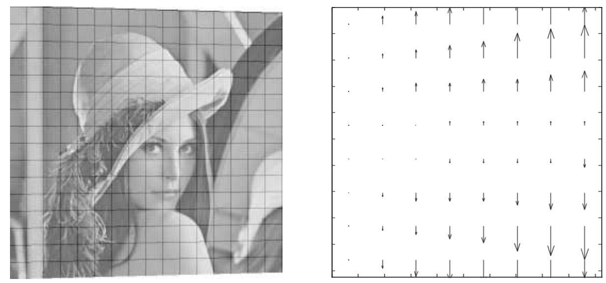
V. EXPERIMENTAL RESULTS

We evaluated the performances of our resynchronization technique against various kind of small geometrical deformations. The tests have been performed on a database of 500 images of size 512×512 pixels. The payload was set to standard value of 64 bits. Figure 17 illustrates some of the geometrical attacks applied to the watermarked image (e.g. rotation, translation, scale, Stirmark, skew, etc.), as well as the way in which they have been compensated by the resynchronization process using block-matching only. In those examples, we can notice that the optical flow fields, although imperfect, mainly follow the geometrical deformations, which subsequently enables to extract all the 64 information bits of the watermark.

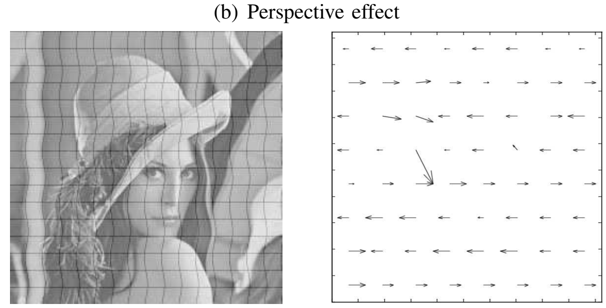
Experimentally, we have noticed that the resynchronization process mainly fails when the watermarked image undergoes large affine transformations (e.g. a rotation greater than 3 degrees, translation greater than the search window size, etc.) or large local geometrical distortions. In these cases, the failure of the resynchronization is due to the fact that the deformations cannot be compensated by the block matching process. On the contrary, a combination of local geometrical distortions and photometric manipulations, such as a small rotation followed by a high JPEG compression, may also locally affect the resynchronization process. Indeed, the watermark is not uniformly embedded into the image and its



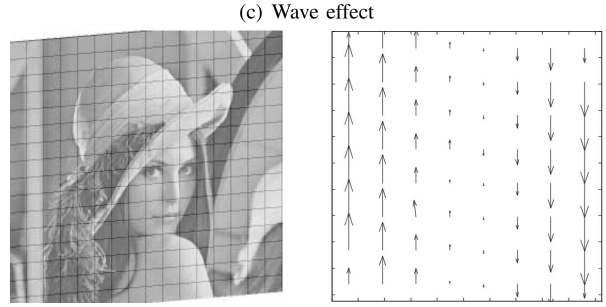
(a) Rotation of 2 degrees



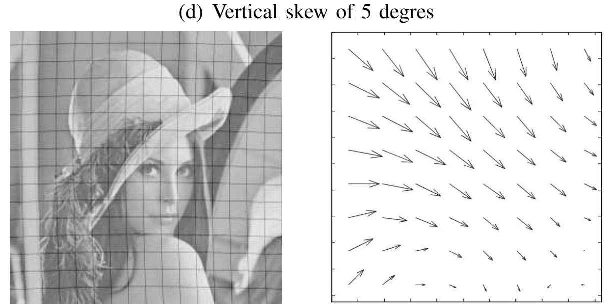
(b) Perspective effect



(c) Wave effect



(d) Vertical skew of 5 degrees



(e) *Stirmark* attack

Fig. 17. Examples of geometrical deformations compensated by the proposed resynchronisation framework.

robustness is different according to the frequential contents of the different parts of the image. Typically, in textured regions the watermark is mainly modulated with high frequencies which makes it invisible, but also less robust to low-pass filtering, bilinear interpolation, etc. Thus the bits which were hidden in those areas, and particularly the reference bits, may be erroneous causing incorrect block pairings. If the incorrect block pairings are scattered they can be easily detected and corrected (or eliminated) using the specific post-processing described in paragraph IV-C.5. Figure 18 depicts for instance how the optical flow obtained by block matching with an image distorted by Stirmark can be improved. First, the rigidity parameter λ enables to correct some few wrong estimated displacements. Second, the multi-scales framework permits to obtain a denser optical flow.

Finally, to obtain a quantitative evaluation of the different proposed resynchronization techniques, we have attacked our database of watermarked images with Stirmark with varying strength [44] and looked at the Message Error Rate (MER). We define the MER as the ratio of the number of erroneous messages on the number of extracted messages. A message is erroneous if at least one of its bits is false. Three alternative resynchronization methods have been investigated, namely no resynchronization, block matching only resynchronization and elastic graph matching resynchronization. The results reported in Figure 19 clearly demonstrates that resynchronization is required since the algorithm is quickly defeated otherwise. Furthermore, the EGM-based regularization process proposed in paragraph IV-C.5 enables to significantly enhance the performances of the block-matching based baseline resynchronization module. For a Stirmark strength equal to 1 (default strength value), it makes the MER drop from 23.21% to 14.64%. Another interesting observation is that, for low Stirmark strength, the BM resynchronization introduces some errors in comparison to the system without any resynchronization. This is due to the fact some estimated displacements are wrong and thus may introduce errors. In fact, empirical results have shown that BM resynchronization lower the robustness performances of the algorithm with respect to photometric attacks e.g. JPEG compression. This issue seems to be less critical with EGM resynchronization since a few wrong displacement estimations can be corrected. However, if the initial optical flow obtained by block-matching contains too much errors, they will not be corrected and performances against photometric attacks wil also be degraded.

VI. Detection of Global Linear Transforms

As underlined the proposed approach is not adapted to compensate for important global transformations. In order to counterbalance some of the shortcomings of the previous resynchronization method, we have developed a complementary method [46] allowing global linear transforms of the image (i.e. rotation, scale, skew, etc.) to be compensated. The main idea of this method consists in embedding a repetitive square

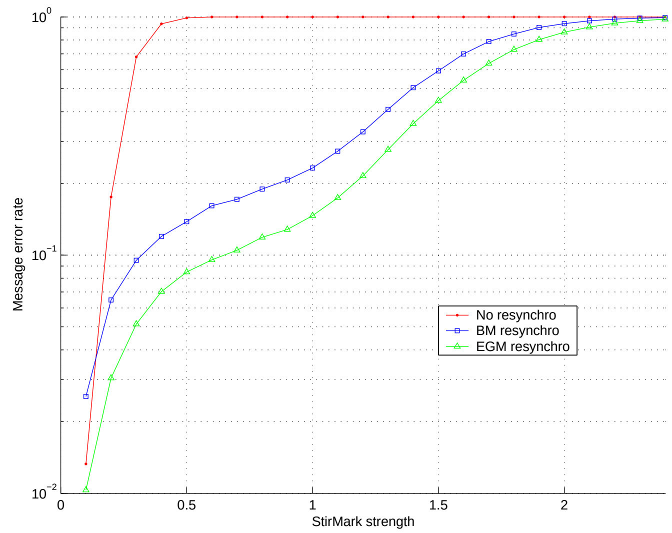


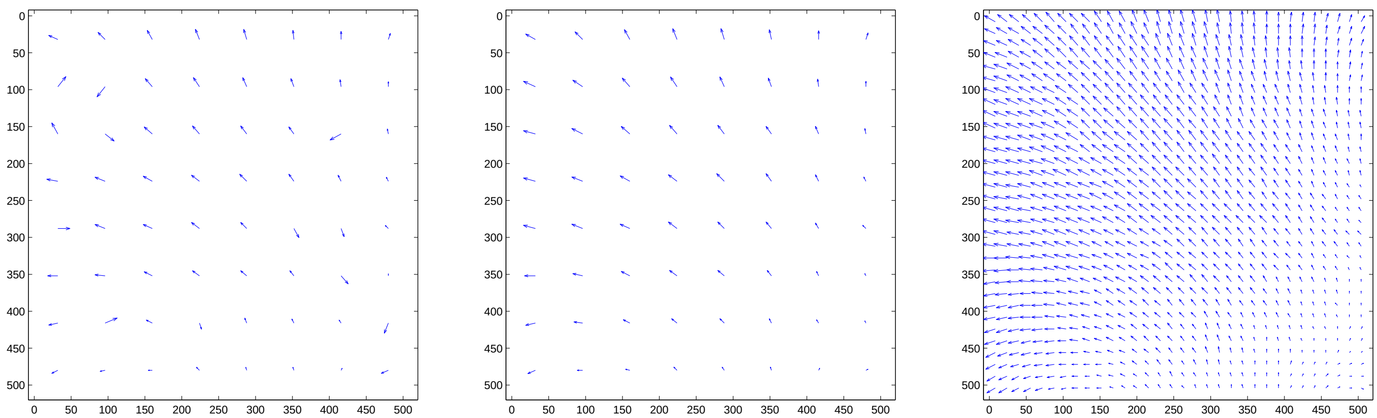
Fig. 19. Message error rate versus the strength of the Stirmark attack for different resynchronization methods.

pattern into the cover[1]. The addition of a periodic pattern in the spatial domain involves the appearance of regularly aligned peaks in the frequential domain. The alignment of those points is kept under any affine transform. Thus the determination of the linear transform A_s applied to the image is realized by estimating the parameters of the dual linear transform A_f into the frequential domain. Indeed, the two affine transforms A_s and A_f are connected by the following relation:

$$A_f = \left(A_s^T\right)^{-1} \tag{11}$$

In this paragraph, we only briefly describe the baseline of the method. The idea is to XOR only the useful bits during the formatting of the watermark in order to generate a periodical pattern from the reference bits and keep secure information bits by keeping a global encryption. Thus the resynchronization process first at all consists in detecting the peaks in the DFT domain applying adaptive thresholding. Then we estimate the two main axes based on Hough transform and the periods from aligned points along these two directions. Knowing the original alignment and period of the peaks, we can deduce the parameters of the undergone affine transform A_s modulo 8 isometries (i.e. identity, horizontal, vertical and central symmetries, $90°$ and $-90°$ rotations, and 2 reflexions). These ambiguities are due to the fact that the aligned peaks do not give any information about the correct vector orientations along the two main directions. To solve this problem, we must test in parallel the 8 possible inverse affine transforms and select the one whose $|b_k|$'s have the highest mean value, where b_k is defined in Equation (6).

[1] As mentioned earlier in the state of the art, this idea was first introduced by Kutter [27]. The approach based on the Hough transform presented hereafter has been done independently, but in parallel with the one proposed by Deguillaume et al. [28].



(a) *Block matching* (b) *Regularization with EGM* (c) *Higher density with multi-scales*

Fig. 18. *Illustration of the iterative regularization of the optical flow when EGM is used. The watermarked image has been submitted to the random bending attack.*

VII. Concluding Remarks

Geometrical transformations, and more precisely local geometrical distortions, are the Achilles heel for many watermarking schemes. Most solutions proposed in the literature only address the problem of affine transforms. In this article, we have described an efficient resynchronization method against slight global and local geometrical deformations such as the ones generated by *Stirmark* attack. The proposed approach consists in adding predefined additional information to the useful message bits at the embedding step. During the extraction step, these bits are then used as anchor points to estimate and compensate for geometrical manipulations; the resynchronization process consisting in matching the reference bits with the extracted cover. In most cases, the method is able to compensate for all the geometrical deformations which can be locally approximated by block-matching. Nevertheless, a combination of geometrical and photometric manipulations such as a small global rotation followed by an average to strong JPEG compression may affect the resynchronization process. However, the proposed resynchronization method reaches its limits when the watermarked image undergoes large geometrical (global or local) distortions or a significant photometric attack. Some of these shortcomings, such as affine transforms, can be counterbalanced using a complementary resynchronization method. However, the difficulty is then to make both approaches work together to be able to cope with any kind of geometric distortion, in particular a combination of global and local geometric attacks.

References

[1] S. Katzenbeisser and F. A. P. Petitcolas, Eds., *Information Hiding: Techniques for Steganography and Digital Watermarking*. Artech House Publishers, 1999.

[2] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*. Morgan Kaufmann Publishers, 2001.

[3] E. Koch and J. Zhao, "Towards robust hidden image copyright labeling," in *IEEE Workshop on Nonlinear Signal and Image Processing*, June 1995, pp. 452–455.

[4] F. Hartung and B. Girod, "Digital watermarking of raw and compressed video," in *Digital Compression Technologies and Systems for Video Communication*, ser. Proceedings of SPIE, vol. 2952, October 1996, pp. 205–215.

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

[6] M. Barni, F. Bartolini, V. Cappelini, A. Lipi, and A. Piva, "A DWT-based technique for spatio-frequency masking of digital signatures," in *Security and Watermarking of Multimedia Contents*, ser. Proceedings of SPIE, vol. 3657, January 1999, pp. 31–39.

[7] J. K. S. F. Hartung and B. Girod, "Spread spectrum watermarking: Malicious attacks and counter-attacks," in *Security and Watermarking of Multimedia Contents*, ser. Proceedings of SPIE, vol. 3657, January 1999, pp. 147–158.

[8] K. Tsang and O. Au, "A review on attacks, problems and weakness of digital watermarking and the pixel reallocation attack," in *Security and Watermarking of Multimedia Contents III*, ser. Proceedings of SPIE, vol. 4314, January 2001, pp. 55–63.

[9] F. Petitcolas, R. Anderson, and M. Kuhn, "Attacks on copyright marking systems," in *Proceedings of the Third International Workshop on Information Hiding*, ser. Lecture Notes in Computer Science, vol. 1525, April 1998, pp. 218–238.

[10] F. Petitcolas and J.-L. Dugelay, "Possible counter-attacks against random geometric distortions," in *Security and Watermarking of Multimedia Contents II*, ser. Proceedings of SPIE, vol. 3971, January 2000, pp. 338–345.

[11] C. Rey, G. Doërr, G. Csurka, and J.-L. Dugelay, "Toward generic image dewatermarking?" in *Proceedings of IEEE International Conference on Image Processing*, vol. 3, September 2002, pp. 633–636.

[12] M. Kutter, F. Jordan, and F. Bossen, "Digital watermarking of color images using amplitude modulation," *Journal of Electronic Imaging*, vol. 7, no. 2, pp. 326–332, April 1998.

[13] 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*, ser. Proceedings of SPIE, vol. 5020, January 2003, pp. 203–214.

[14] P. Dong, J. Brankov, N. Galatsanos, and Y. Yang, "Geometric robust watermarking through mesh model based correction," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 3, September 2002, pp. 493–496.

[15] F. Davoine, P. Bas, P.-A. Hebert, and J.-M. Chassery, "Watermarking et résistance aux déformations géométriques," in *Proceedings of Compression et Représentation des Signaux Audiovisuels*, June 1999.

[16] N. Johnson, Z. Duric, and S. Jajodia, "Recovery of watermarks from distorted images," in *Proceedings of the Third International Workshop on Information Hiding*, ser. Lecture Notes in Computer Science, vol. 1768, September 1999, pp. 318–332.

[17] D. Coltuc and P. Bolon, "Robust watermarking by histogram specification," in *Proceedings of the IEEE International Conference of Image Processing*, vol. 2, October 1999, pp. 236–239.

[18] J. J. K. Ó Ruanaidh and T. Pun, "Rotation, scale and translation invariant digital image watermarking," *Signal Processing*, vol. 66, no. 3, pp. 303–317, May 1998.

[19] C. Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, M. L. Miller, and Y. M. Lui, "Rotation, scale, and translation-resilient public watermarking for

images," in *Security and Watermarking of Multimedia Contents II*, ser. Proceedings of SPIE, vol. 3971, January 2000, pp. 90–98.

[20] F. Lin and R. D. Brandt, "Towards absolute invariants of images under translation, rotation and dilatation," *Pattern Recognition Letters*, vol. 14, no. 5, pp. 369–379, May 1993.

[21] M. Alghoniemy and A. H. Tewfik, "Geometric distortion correction through image normalization," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, vol. 3, August 2000, pp. 1291–1294.

[22] V. Solachidis and I. Pitas, "Self-similar ring shaped watermark embedding in 2-D DFT domain," in *Proceedings of European Signal Processing Conference*, vol. 4, September 2000.

[23] D. Fleet and D. Heeger, "Embedding invisible information in color images," in *Proceedings IEEE International Conference on Image Processing*, vol. 1, October 1997, pp. 532–535.

[24] A. Tirkel, C. Osborne, and T. Hall, "Image and watermark registration," *Signal Processing (Special Issue on Watermarking)*, vol. 66, no. 3, pp. 377–384, May 1998.

[25] D. Gruhl and W. Bender, "Affine invariance," http://nif.www.media.mit.edu/DataHiding/affine/affine.html, 1995.

[26] S. Pereira and T. Pun, "Fast robust template matching for affine resistant image watermarking," in *Proceedings of the Third International Workshop on Information Hiding*, ser. Lecture Notes in Computer Science, vol. 1768, September 1999, pp. 199–210.

[27] M. Kutter, "Watermarking resisting to translation, rotation and scaling," in *Multimedia Systems and Applications*, ser. Proccedings of SPIE, vol. 3528, November 1998, pp. 423–431.

[28] F. Dehuillaume, S. Voloshynovskiy, and T. Pun, "A method for the estimation and recovering from general affine transforms in digital watermarking applications," in *Security and Watermarking of Multimedia Contents IV*, ser. Proceedings of SPIE, vol. 4675, January 2002, pp. 313–322.

[29] S. Voloshynovskiy, F. Deguillaume, and T. Pun, "Multibit digital watermarking robust against local non linear geometrical distortions," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 3, October 2001, pp. 999–1002.

[30] S. Voloshynovskiy, A. Herrigel, and Y. B. Rystar, "Watermark template attack," in *Security and Watermarking of Multimedia Contents III*, ser. Proceedings of SPIE, vol. 4314, January 2001, pp. 394–405.

[31] D. Delannay and B. Macq, "A method for hiding synchronization marks in scale and rotation resilient watermarking schemes," in *Security and Watermarking of Multimedia Contents IV*, ser. Proceedings of SPIE, vol. 4675, January 2002, pp. 548–554.

[32] DIGIMARC Corp., "http://www.digimarc.com."

[33] M. Kutter, S. K. Bhattacharjee, and T. Ebrahimi, "Towards second generation watermarking schemes," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 1, October 1999, pp. 320–323.

[34] P. Bas, J.-M. Chassery, and B. Macq, "Geometrically invariant watermarking using feature points," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 1014–1028, September 2002.

[35] C. Harris and M. Stephen, "A combined corner and edge detector," in *Proceedings of the Fourth Alvey Vision Conference*, vol. 15, 1988, pp. 147–151.

[36] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Transactions on Information Theory*, vol. 44, no. 5, pp. 1897–1905, September 1998.

[37] A. Nikolaidis and I. Pitas, "Robust watermarking of facial images based on salient geometric pattern matching," *IEEE Transactions on Multimedia*, vol. 2, no. 3, pp. 172–184, September 2000.

[38] J.-L. Dugelay and S. Roche, "Method for hiding binary data in a digital image," Pending Patents PCT/FR99/00485 (EURECOM 09-PCT), March 1999.

[39] J.-L. Dugelay and C. Rey, "Method of marking a multimedia document having improved robustness," Pending Patent EUP99480075.3 (EURECOM 14 EP), May 2001.

[40] Y. Fisher, *Fractal Image Compression: Theory and Applications*. Springer-Verlag, 1994.

[41] C. Rey, K. Amis, J.-L. Dugelay, R. Pyndiah, and A. Picart, "Enhanced robustness in image watermarking using block turbo codes," in *Security and Watermarking of Multimedia Contents V*, ser. Proceedings of SPIE, vol. 5020, January 2003, pp. 330–336.

[42] J. R. Jain and A. K. Jain, "Displacement measurement and its application in the interframe image coding," *IEEE Transaction on Communications*, vol. 29, no. 12, pp. 1799–1808, December 1981.

[43] D. Delannay, I. Setyawan, R. Lagendijk, and B. Macq, "Relevant modeling and comparison of geometric distortions in watermarking systems," in *Application of Digital Image Processing XXV*, ser. Proceedings of SPIE, vol. 4790, July 2002, pp. 200–210.

[44] G. Doërr, C. Rey, and J.-L. Dugelay, "Watermark resynchronization based on elastic graph matching," in *Proceedings of the International Conference on Sciences of Electronic, Technologies of Information and Telecommunications*, March 2005.

[45] M. Lades, J. Vorbrüggen, J. Buhnmann, J. Lange, C. Malsburg, R. Würtz, and W. Konen, "Distortion invariant object recognition in the dynamic link architecture," *IEEE Transactions on Computers*, vol. 42, no. 3, pp. 300–311, March 1993.

[46] C. Rey, "Image watermarking : Enhanced robustness and image authentication (in french)," Ph.D. dissertation, Université d'Avignon et des Pays de Vaucluse, France, February 2003.

Jean-Luc Dugelay (Ph.D. 92, IEEE M'94-SM'02) joined the Eurécom Institute (Sophia Antipolis, France) in 1992. He is currently a Professor in the Department of Multimedia Communications and is in charge of the Image and Video Group for Multimedia Communications and Applications. His research interests include security imaging (watermarking and biometrics), image/video coding, facial image analysis, face cloning and talking heads. He has published over 80 technical papers and holds three international patents. He has given several tutorials on digital watermarking co-authored with F. Petitcolas (Microsoft Research, Cambridge, England). In addition to national French projects, his group is involved in the European Network of Excellence *E-Crypt*. He is also serving as a Consultant in digital watermarking for France Télécom R&D and STMicroelectronics. He is an Associate Editor for the IEEE Transactions on Multimedia, the IEEE Transactions on Image Processing, the EURASIP Journal on Applied Signal Processing and the Kluwer Multimedia Tools and Applications.

Stéphane Roche obtained his Ph.D. in 1999 from ENST Paris on digital watermarking with highest honors. He also contributed to the first book on watermarking (information hiding techniques for steganography and digital watermarking, Artech House 1999).

Christian Rey received a Master's degree in Computer Science and Artificial Intelligence from the University of Sciences of Luminy (Marseille), France in 1998. He joined Eurecom in May 1999 as Ph.D. Student. He received his Ph.D. degree from the University of Avignon in 2003. He was involved in the French national research project RNRT AQUAMARS, on workpackage image authentication. He is the coauthor of several publications, technical demos, and patents related to watermarking.

Gwenaël Doërr received in 2001 the telecommunications engineering degree from Institut National des Télécommunications (Télécom INT), Evry, France and the M.S. degree in computer science from Université de Nice Sophia-Antipolis (UNSA), Sophia-Antipolis, France. He was an intern at NEC Research Institute, Princeton, NJ from April to September 2001, winning the Louis Leprince Ringuet Award for his work on dirty paper trellis watermarks. He then obtained his Ph.D. in 2005 from Université de Nice Sophia-Antipolis on security issue and collusion attacks in video watermarking. He joined Adastral Park Postgraduate Campus as a Lecturer in mid-August 2005.