Influence of a Large Image Watermarking Scheme Parallelization on Possible Attacks

Oscar Divorra Escoda[1], Rosa Maria Figueras i Ventura[1], Eric Debes[2], Touradj Ebrahimi[1]

[1] Signal Processing Laboratory, Swiss Federal Institute of Technology (EPFL), CH-1015 Lausanne, Switzerland

[2] Microprocessor Research Labs, Intel Corporation, 2200 Mission College Blvd. Santa Clara, CA 95052

ABSTRACT

Digital data representation provides an efficient and fast way to access to information and to exchange it. In many situations though ownership or copyright protection mechanisms are desired. For still images and video, one possible way to achieve this is through watermarking. Watermarking consists of an imperceptible information embedded within a given media. Parallel Processing Watermarking Embedding Schemes have demonstrated to be efficient from a computational and memory usage point of view for very large images. These schemes consist in dividing the image into tiles and watermarking each independently. The processing allows the use of a parallel computation scheme. The watermarking method used in the scope of this work is a parallel variant of an approach known as self-referenced Spread Spectrum signature pattern. Since the watermarking scheme has been modified through tiling, the extra references due to signature replication can be used in the retrieval. This work describes the above mentioned approach to watermark images and provides an analysis of its performance.

Keywords: Watermarking, Large images, Multi-threading, Memory Requirements, Watermark retrieval, Watermark extraction, Attacks, Wiener Filter

1. INTRODUCTION

Because of the digital revolution combined with widespread network infrastructures, the protection of digital data such as images or video is becoming necessary. One protection technique for still images, called watermarking, consists in hiding in the image an information invisible to the human eye.

Recently, many watermarking schemes have been proposed to sign digital images. However, most of these schemes are not well suited for large images because they are very often applied to the complete image at once. This means that the memory consumption of these algorithms[1] is very high when the size of the image increases. Moreover, such schemes cannot take advantage of more than one processor, even though current PC-based servers integrate very often more than one. Therefore existing schemes need to be adapted to overcome these shortcomings.

A possible parallelization scheme is proposed by Debes *et al.*[1] This method consists in tiling the image into sub-images. An existing watermarking scheme[2–5] is then applied to each sub-image. This kind of parallelization scheme introduces more redundancy in the watermark signal, which has two main effects: on one hand it improves the robustness of the watermarking and on the other hand the redundancy can be used against attacks.

In this paper, the influence of this parallelization scheme on the robustness and on possible attacks is studied. In the next section, the parallelization technique for large image watermarking is briefly reviewed. In section 3, the influence of the parallelization technique on the watermark is studied. Moreover a possible attack using a Wiener filter is proposed, taking advantage of the redundancy in the watermark signal. Then standard geometric attacks are studied in section 4 for images watermarked with the parallelized scheme to show that the tiling improves the robustness of the watermark. Finally section 5 concludes the paper.

Further author information:
Oscar Divorra Escoda: Email: Oscar.Divorra@epfl.ch, Rosa Maria Figueras i Ventura: Email: Rosa.Figueras@epfl.ch, Eric Debes: Email: debes@ieee.org, Touradj Ebrahimi: E-mail: Touradj.Ebrahimi@epfl.ch

2. LARGE IMAGE WATERMARKING THROUGH TILING

2.1. Methodology

In this section, we briefly review the process of watermarking large images through tiling which is was first proposed by Debes *et al.*[1] . The main goal of this method is to reduce the memory consumption of watermarking process applied to very large images in parallel[6] in current generation of PC workstations without using virtual memory ("disk swapping"). This can be very useful for example in a client-server setup, in which clients submit images to be watermarked to a server which processes the images and sends them back to the client PC. The second aim of the method is to take advantage of the multiple processors available in most servers.

One thread is created[7] to watermark each tile and thus the computational intensive processing tasks are automatically distributed on each processor by the operating system. In addition, in order to parallelize the input/output operations, additional threads are created to read and write the tiles so that the current tile is watermarked while the previous one is written and the next one is read.

2.2. Choice of a tile size

The first important choice is the tile size. It should be large enough so that the thread management overhead can be neglected compared to the processing time. On the other hand, it should be small enough in order to cope with memory requirements. Finally the tile size should be large enough to efficiently embed the signature with the considered watermarking scheme [2–5] . An experimental study[1] has shown that 256×256 is a good compromise. Furthermore, it presents the advantage of allowing a few tiles to be stored in the L2 cache of PC Workstations.

2.3. Choice of the number of threads

The second choice concerns the number of threads to be created. As explained by Debes et al. [1] , there are always $3N$ threads running: N threads reading the next N tiles to be processed, N threads watermarking the current tiles and N threads writing (or sending to the net) the watermarked tiles back to the disk (or to the remote client). An experimental study [1] has shown that 3 times 6 threads is a good compromise for a dual CPU machine.

2.4. Visual Quality

The quality of the watermarked image is, of course, very important for the final user and no compromise should be made on this issue when parallelizing the watermarking scheme. Visual quality tests performed by different users with our technique revealed that parallelization of the watermarking scheme does not modify the quality of the watermarked image. Indeed, the energy of the watermark embedded in each tile can be adapted so that the overall quality remains constant in the whole image. In fact the tiling method provides more freedom to distribute the energy of the watermark in the image and makes it possible to embed a more robust signature in tiles where it will be less visible.

3. INFLUENCE OF TILING ON THE WATERMARKING

3.1. Autocorrelation

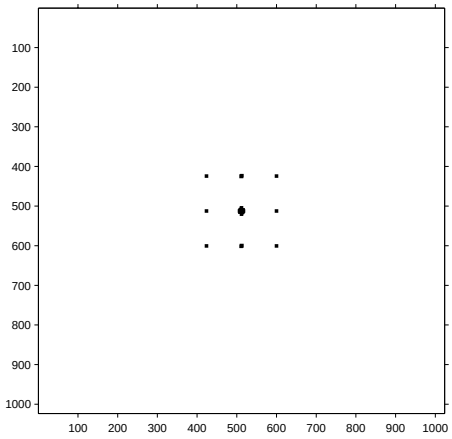
To fully understand the importance of the autocorrelation in the watermarking detection, the original watermarking scheme used here should be briefly described. To have a watermark resistant to scaling and rotations, Kutter [2] proposed to embed four shifted interlaced spread spectrum signatures in the same image. This gives a predefined pattern in the autocorrelation of the watermark, consisting of a central peak surrounded by eight smaller peaks (see Fig. 1).

The watermark autocorrelation pattern can be used to detect if the image has suffered any kind of geometric attack. When this is the case, the autocorrelation pattern of the watermark inserted in the image will be modified by the attack. Thanks to this, possible image attacks will be detected, and the inverse transform computed. The detection and retrieval of the spread spectrum signal is then performed on the inverse transformed image. Due to the importance of the watermark autocorrelation pattern in the signature detection, a detailed analysis of the parallelized scheme and its consequences in the autocorrelation of the image is needed.

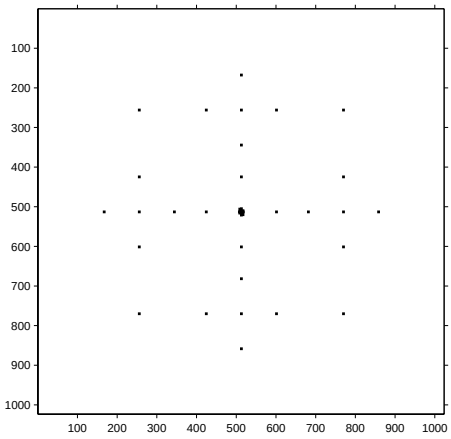
When the parallelization scheme is used in the watermark embedding, the watermarked image will take the form:

$$\hat{C} = C + \sum_i \sum_j \alpha_{[i,j]}[n,m] \cdot w[n - i \cdot T_n,\ m - j \cdot T_m], \tag{1}$$

(a) Watermarked image



(b) Watermark autocorrelation (only one tile for the whole image)



(c) Watermark autocorrelation (Image divided in four tiles)

Figure 1. Autocorrelation of the watermark estimation in an image without tiling and with tiling. The watermark has been estimated by adaptively filtering (with a Wiener filter[8]) the image.

where C represents the original image, w the signature pseudo-random sequence, T_n and T_m the tile dimensions (periodicity intervals of w) and α the weighting function which takes into account the Human Visual System and the eye sensitivity to contrast and noise visibility.[2,9] The watermark signal (double sum in Eq. 1) is a modulation of the spread-spectrum basis, which can be considered as a Stationary Stochastic Process of zero mean[10]:

$$w[n,m] = \sum_i b_i \cdot s_i[n,m] = \sum_i b_i \cdot s_i(n-\delta_i^x, m-\delta_i^y), \tag{2}$$

where b_i is the signature bit value and s_i is the spread spectrum signal.

The watermark autocorrelation is shift invariant[11], so the autocorrelation function will only depend on the distance, not on the position. As a consequence of shift invariance, the resulting autocorrelation is a periodic version of the simple watermark, with periods T_n and T_m (where the tile size is $T_m \times T_n$):

$$R_{ww_{periodic}}[n,m] = \sum_{l,k}\sum_{i,j} w[l-i\cdot T_n, k-j\cdot T_m] \cdot \sum_{i',j'} w[l-n-i'\cdot T_n, k-m-j'\cdot T_m] =$$

$$= \sum_{i,j} \sum_{i',j'} R_{ww}[n-(i'-i)\cdot T_n, m-(j'-j)\cdot T_m] = \Big|_{\substack{(i'-i)=t \\ (j'-j)=t'}} \sum_{t,t'} R_{ww}[n-t\cdot T_n, m-t'\cdot T_m]. \quad (3)$$

In the above expression the effect of α has not been taken into account. Below we will show that the value α affects the amplitude of the peaks only, and does not modify their positions.

The weighting function α is a variable that depends on the image itself. It can be considered as a random variable independent from the spread spectrum signal.

The autocorrelation to compute is now:

$$R_{w'w'}[l,k] = E\{(\alpha[n,m]\cdot w[n,m])\cdot(\alpha[n-l,m-k]\cdot w[n-l,m-k])\}. \quad (4)$$

Since $w[n,m]$ is the spread spectrum sequence and can be considered uncorrelated with the image (and so with α), the final expression of the estimated watermark autocorrelation will be:

$$R_{w'w'}[l,k] = R_{ww}[l,k]R_{\alpha\alpha}[l,k]. \quad (5)$$

The above expression comes from the fact that, when considering two Gaussian variables x, y:

$$E\{x(u)y(u+\tau)x(v)y(v+\tau)\} = R^2_{xy}(\tau) + R_x(v-u)R_y(v-u+\tau) + R_{xy}(v-u+\tau)R_{yx}(v-u-\tau). \quad (6)$$

Extending expression (6) to our case, we consider $R_{xy}(v-u) = 0 \ \forall \ u,v$ since x (the pseudo-random sequence) and y (the weighting function) are uncorrelated and $R_x(v-u) \approx 0 \ \forall \ v \neq u$ since x is an uncorrelated pseudo-random sequence. Eq. (6) can then be approximated by:

$$E\{x(u)y(u+\tau)x(v)y(v+\tau)\} = R_x(v-u)R_y(v-u+\tau). \quad (7)$$

In the watermark detection algorithm, the most important fact is the autocorrelation peak location. If the peak position is not affected by the weighting function, its effect can be omitted to a large extent.

As α varies slowly with the changes of the image compared to the spread-spectrum embedded sequence ($w[n,m]$), it can be considered to be approximately constant in a local area of the image.[2–4] The Fourier transform of a constant is a Dirac delta δ. So α, in the Fourier domain, can be approximated by a Dirac delta function $F\{\alpha\} \simeq K\delta(f)$. Since the Noise Visibility Function and the weighting function are never negative, α cannot be placed at any frequency, it has to be centered at zero frequency. Taking the Fourier transform of Eq. 4:

$$S_{w'w'} = F\{R_{w'w'}\} = S_{ww} * S_{\alpha\alpha}, \quad (8)$$

where S indicates the spectral density. From the above expression, it can be deduced that the effect of the α function is just a scaling factor in the amplitude. As:

$$R_{w'w'}[n,m] = F^{-1}\{(F\{\alpha\} * F\{w\})\cdot(F\{\alpha\} * F\{w\})\}, \quad (9)$$

the following approximation holds:

$$R_{w'w'}[n,m] \approx F^{-1}\{KF\{w\}\cdot KF\{w\}\} = K^2 R_{ww} \Longrightarrow R_{w'w'} \approx K^2 R_{ww}. \quad (10)$$

As the peak position is not affected, the effect of the weighting function in the watermark autocorrelation will not be taken into account from now on.

As have been seen, the fact of tiling causes a periodization with a tile-size periodicity value, and because of the properties of the watermark autocorrelation, of the watermark autocorrelation as well:

$$R_{ww_{periodic}}[n,m] = \sum_{t,t'} R_{ww}[n-t\cdot T_n, m-t'\cdot T_m]. \quad (11)$$

This fact will be extremely important when looking at attacks in the watermarked image.

3.2. Problem introduced by the tiling

Tiling modifies the autocorrelation pattern of the image, so it is necessary to adapt the watermark detector to that. This is done by adding the possibility of finding more than four surrounding autocorrelation peaks and a central peak, and detecting a central peak and a certain number of secondary peaks. It was found that detecting two peaks for each axis (where the axis is defined by the strongest secondary peak found) is sufficient for retrieval of watermarks after any geometric transform.

Another problem introduced by the tiling scheme (a problem that can turn out to be an advantage for the detection, as will be seen in section 4) is that the extra redundancy added can be used by a hacker for removal or extraction of the watermark from the image. This kind of attack is explained in more details in the next section.

3.3. Malicious Signature Removal through filtering

The information given by the pattern obtained in Sec. 3.1 could be used for malicious attacks. Thanks to the correlation peaks, a precise localization of the signature tiles is possible. This, at first sight, may seem not so important because only the position is known and there is no knowledge about the signature itself. But this could be turned into an attack, specially in large images if a fixed tile size is used independently of the image size. This implies that the bigger the image, the larger the number of tiles. In addition many tiles could contain rather uniform regions. This is represented in Fig. 2, where a relatively large (2048x2560) natural image is divided into tiles of size 256x256 pixels. It follows from Fig. 2 that several tiles correspond to very uniform objects or surfaces, such as those tiles corresponding to the sky. In general, this kind of areas will be those corresponding to the background. Spread-Spectrum watermarking corresponds to the addition of a watermark signal to the original image. In tiles where the information of the image is merely reduced to a uniform area, they will represent the signature pattern with an offset level corresponding to the luminance level of the uniform region. From that, the signature pattern can be quite well separated from the image, without the inconvenient appearance of object edges due to the filtering process. The watermark pattern is a

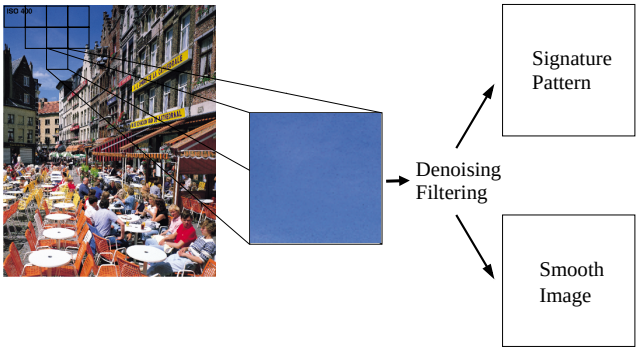


Figure 2. Tiling scheme

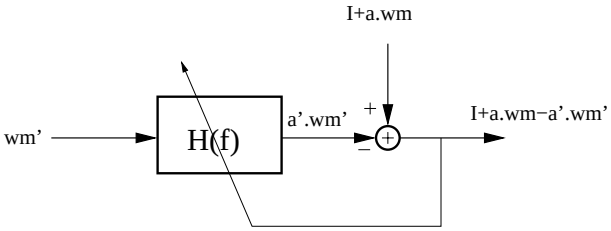


Figure 3. Interference removing wiener filter use.

2D noise-like signal. From that, it is assumed that the image and the pattern are uncorrelated.

According to Eq.(1), the watermark signal can be considered as an interference signal.

Since the watermark pattern basis is known, an interference removal could be performed using a classical approach such as Wiener filtering[12] .

From Eq. (1)

$$I'[n,m] = \hat{C} - h[n,m] \cdot wm', \tag{12}$$

where I' is the estimated clean image, $\hat{C}$ the watermarked image, wm' is the estimated watermark pattern, and $h[n,m]$ represents the filter that will pre-process the estimated watermark pattern wm' to adapt it to the local amplitude of the watermark.

The objective here is to find the best $h[n,m]$ coefficients that would allow to eliminate as much as possible the watermark. One approach is to select $h[n,m]$ so that the variance of I' is minimized over a specified neighborhood for every point $[n,m]$.

Considering the variance on a local neighborhood of size $[2N+1] \times [2M+1]$

$$\sigma^2_{[n,m]} = \frac{1}{(2N+1)(2N+1)} \sum_{l=-N}^{N} \sum_{t=-M}^{M} \left[I'[n+l,m+t] - E\{I'[n,m]\} \right]^2. \tag{13}$$

From Eq. 12, Eq. 13 and considering that $h[n,m]$ is constant all over the neighborhood, it follows:

$$\sigma^2_{[n,m]} = \frac{1}{(2N+1)(2M+1)} \sum_{l=-N}^{N} \sum_{t=-M}^{M} \{ [\hat{C}[n,m] - h[n,m] \cdot wm'[n+l][m+t]] - [E\{\hat{C}[n,m]\} - h[n,m] \cdot E\{wm'[n,m]\}] \}^2. \quad (14)$$

To minimize the presence of the interfering signal, $\sigma^2_{[n,m]}$ has to be minimized. The following should be satisfied:

$$\frac{\partial \sigma_[n,m]^2}{\partial h[n,m]} = 0, \quad (15)$$

resulting in:

$$h[n,m] = \frac{E\{\hat{C}[n,m] \cdot wm'[n,m]\} - E\{\hat{C}[n,m]\} \cdot E\{wm'[n,m]\}\}}{E\{wm'[n,m]^2\} - E\{wm'[n,m]\}^2} \quad (16)$$

The embedded signature pattern is weighted by a function to improve watermark's invisibility. Because of that, the assumption of stationarity when using the wiener filter fails. To overcome this fact a local evaluation of the cross-correlation between the pattern and the watermarked image is necessary. In this way a local estimation of the weighting function is performed using a local window.

The above approach successfully reduces the watermark inserted to an image leading to failure in the detection process. However some visual distortions appear in the surroundings of edges and very textured areas. This is due to two main reasons:

- The additive noise from the image that is mixed with the watermark signature is increased in the process. Since the weighting function of the watermark takes larger values on edges and textures.
- The use of a local window assumes a uniform weighting value inside that window. This is a good assumption in low varying regions, but it fails when the estimation window goes through an edge, failing consequently on the local watermark power estimation.

In order to improve the quality of the resulting image as well as the performance of the "image cleaning", it could be recommendable to take into account edges or regions when computing the neighborhoods. In this way, the problem of distortion around edges would be reduced.

The attack explained above is based on the assumption that the same pattern is used in all the tiles in which the image is divided. A possible counter attack could be the use of at least two keys in the spread-spectrum sequence generation, in order to have two different patterns embedded in the image. In this way, it could be possible to watermark plain (or mostly plain) tiles with one of the two patterns while the remaining tiles would be marked with the second pattern. Such approach would require further changes in the actual retrieval algorithm (discussed in the following section) but this is out of the scope of this work.

4. GEOMETRIC ATTACKS AND PROPOSED SOLUTIONS

4.1. Cropping

The fact of introducing several times the same watermark signature in an image increases the robustness against cropping. Since several copies of the same signature can be found in the image, the percentage of image area needed to detect the watermark is reduced. As cropping is a space windowing:

$$w_{crop}[m.n] = w[m,n] \cdot \Pi \left[\frac{m}{M}, \frac{n}{N} \right], \quad (17)$$

(where Π is the square window) its effect on the watermark autocorrelation is just an attenuation of the autocorrelation peaks, but not a deformation or a displacement:

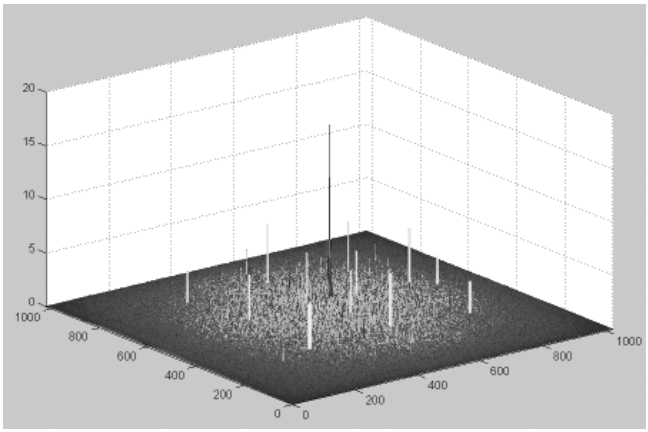
$$R_{ww_{crop}} = R_{ww}[m,n] \cdot R_{\Pi\Pi}[m.n] = R_{ww}[m,n] \Lambda \left[\frac{m}{2M}, \frac{n}{2N} \right], \quad (18)$$

where Λ is the triangular window, resulting of the autocorrelation of the square window.

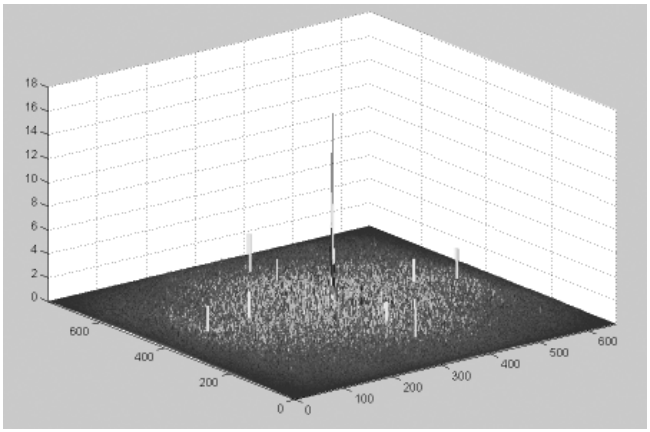
When an image has been cropped, nothing can be done to invert it. Yet, it is highly probable that there will be at least one tile kept intact. Indeed, if no tile is kept intact when cropping a big image, the cropped image will be too small and thus not worth to protect through watermarking. However, even with only a partial tile, detection may still be possible. As the original watermarking scheme[2] is robust to cropping, it will also be the case for the watermarking scheme with tiling. [1]

(a) Original watermarked Lenna

(b) Cropped Lenna



(c) Correlation of the non-cropped image estimated watermark



(d) Correlation of the cropped image estimated watermark

Figure 4. Effect of cropping in the autocorrelation

4.2. Geometric Attacks: Scaling and Rotation

Images are very likely to be affected by transformations such as scaling and rotation. [4] Such transforms can appear when an image is printed and re-scanned again. The current use of a watermarked image by a non-allowed person can include this kind of changes.

The use of a parallelized version of the spread-spectrum watermarking algorithm brings supplementary information for signature retrieval. Scaled or rotated images are likely to introduce distortion peaks in the autocorrelation image. The existence of additional autocorrelation peaks due to signature periodicity increases the chances to detect the correct size of the signature tile.

4.2.1. Scaling

Scaling can be detected through the autocorrelation from the estimated watermark. The autocorrelation gives a pattern formed by a main central peak in every tile surrounded by eight secondary peaks (product of the self-similarity on the watermark pattern). These secondary peaks have a pre-defined distance to the central peak chosen when watermarking the image. If this

distance is larger or smaller than the expected value, this means a scaling of the image has been performed (see Fig. 5). The scaling is proportional to the relation between the original and the observed distance. Considering a scaled watermark:

$$w'[m',n'] = w\left[\frac{m}{a},\frac{n}{b}\right], \tag{19}$$

the inversely scaled Fourier transform is:

$$W'[k',l'] = W[ak,bl], \tag{20}$$

where $W[k,l]$ is the Fourier transform of $w[m,n]$ and $W'[k',l']$ is the Fourier transform of $w'[m',n']$. The weighting factor α is not taken into account because, as shown in section 3.1 it does not affect the peak position. The autocorrelation in the Fourier domain is the spectral density function, which can be computed as the quadratic modulus of the Fourier transform of the signal:

$$S_{w'w'} = |W'[k',l']|^2 = |W[ak,bl]|^2. \tag{21}$$

So, the transform to the spatial domain is:

$$R_{w'w'}[m',n'] = F^{-1}\{S_{w'w'}[k',l']\} = F^{-1}\{S_{ww}[ak,bl]\} = R_{ww}[\frac{m}{a},\frac{n}{b}], \tag{22}$$

which is nothing else than the scaled version of $R_{ww}[m,n]$ with exactly the same scaling applied to the image. As the pre-defined distance between peaks is known, it will be possible to invert the attack and detect the watermark as usual on the basis of what is described above.

4.2.2. Rotation

Rotations can also be detected through the autocorrelation. As the Fourier transform of a rotated signal is its rotated Fourier transform;

$$F\{w[r_\theta\vec{n}]\} = W[r_\theta\vec{k}], \tag{23}$$

where $W[\vec{k}]$ is the Fourier transform of $w[\vec{n}]$ and r_θ is the rotation matrix. If $w'[\vec{n}] = w[r_\theta\vec{n}]$, we will have, as in the previous case, that:

$$R_{w'w'}[\vec{n'}] = F^{-1}\{S_{w'w'}[\vec{k'}]\} = F^{-1}\{S_{ww}[r_\theta\vec{k}]\} = R_{ww}[r_\theta\vec{n}]. \tag{24}$$

According to the expression above, the peaks will also be rotated. With this property the attack will be detected, and it will be possible to invert it. In Fig. 6 an example of rotation in the autocorrelation peaks when the image is rotated is seen (all the peaks suffer the same rotation angle, with respect to the central peak).

4.3. Combination of Attacks

As have been seen in this section, geometric attacks are associated to a transformation matrix. A combination of attacks is nothing else than a product of transformation matrix. Apart from the case of cropping, where there is loss of information, any combination of geometric attacks can be detected through the autocorrelation peaks. With their position the attack matrix can be found, and consequently the inverse transformation matrix computed. The conclusion is that with the autocorrelation peaks information, any geometric attack or combination of them can be detected and inverted. The only condition is that of previous knowledge of the distance between peak positions in the watermark image before the attack (parameter that is intrinsic to the signing algorithm), or any other equivalent information.

The conclusion is that the fact of watermarking through tiling gives more robustness to attacks, thanks to the fact that if the autocorrelation peaks surrounding the central tile peak are too weak and cannot be detected because of added noise, the tile peaks will possibly be present.

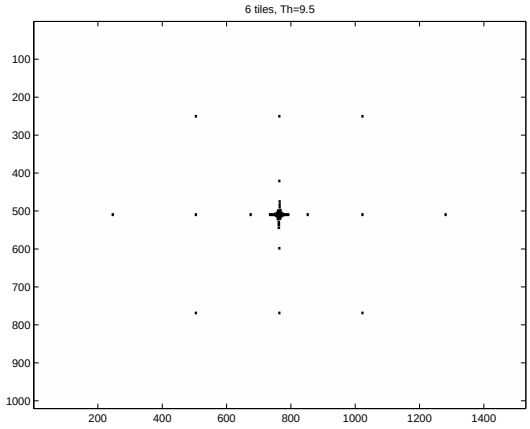
4.4. Computation of the attack inversion matrix

As said before, attack detection is done through autocorrelation. In the autocorrelation the algorithm will search for a central peak and the second most powerful peak. Once this second peak is found, a third peak in the axis formed by central peak and secondary peak will be searched for.

As two different peaks per axis have been detected (the minimum number in order to be able to know whether they are central or secondary peaks) there could exist 16 types of cases (four different cases per axis) of peak identity. On the basis of the possible geometric attacks discussed in sect. 4, the transformation suffered by the image will be assumed as follows:

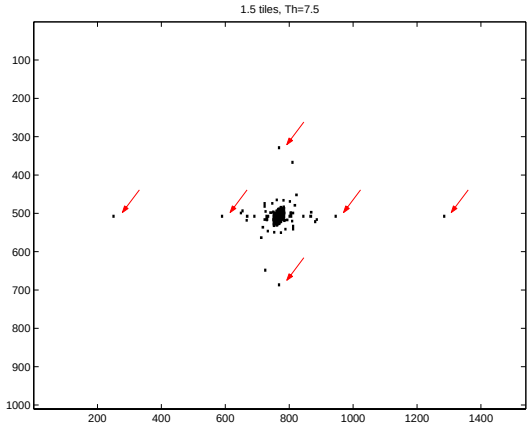
$$y = \mathbf{A}x + \mathbf{B}, \tag{25}$$

(a) Original image



(b) Autocorrelation of the estimated watermark of the original image

(c) Cropped and Scaled Image



(d) Autocorrelation of the cropped and scaled estimated image watermark.

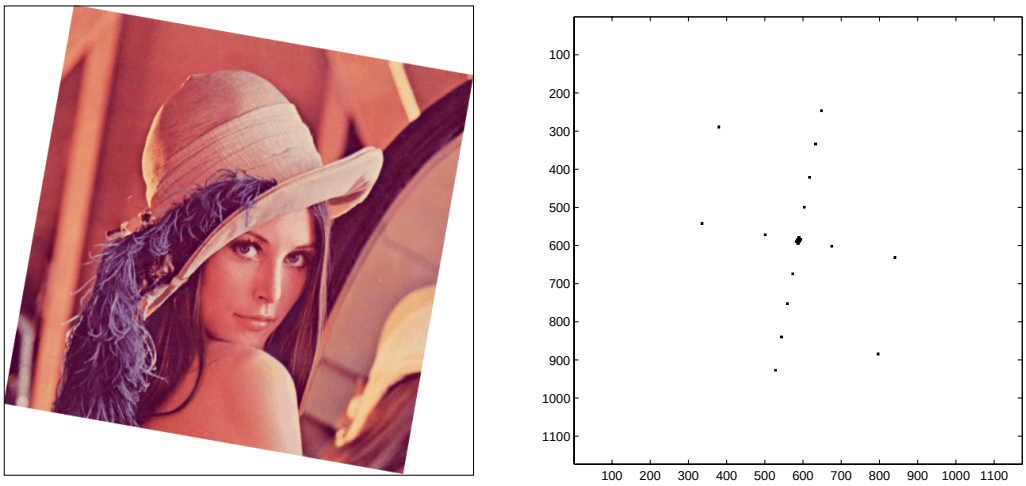
Figure 5. Effect of the scaling in the autocorrelation. In (b) the autocorrelation of the original watermarked image is found and in (d) the autocorrelation of the same image cropped at half size and scaled by 2. Some little peaks appear around the main peak due to distortion introduced by scaling. Arrows indicate which peaks will give the information about the geometrical attack suffered by the image

where **A** and **B** are matrices. The attack matrix **A** will be estimated, through the peak position according to the following formula:

$$\mathbf{A} = \begin{pmatrix} \frac{x_2}{pre\_defined\_distance\_peak_2} & \frac{x_1}{-pre\_defined\_distance\_peak_1} \\ \frac{y_2}{pre\_defined\_distance\_peak_2} & \frac{y_1}{-pre\_defined\_distance\_peak_1} \end{pmatrix}, \tag{26}$$

where (x_1, y_1) and (x_2, y_2) are relative coordinates of two different peaks, one from each axis, from the possibly attacked image. The values corresponding to the pre-defined distances are those corresponding to the distance of the original peak respect to the center of the image when no attack has been performed. This is such that: $pre\_defined\_distance\_peak_2$ is the coordinate value of the horizontal axis original peak, and the $pre\_defined\_distance\_peak_1$ is the coordinate value of the vertical axis original peak (See Fig. 1).

Supposing **A** is an invertible matrix, the transformations the image has suffered will be invertible by applying its inversion matrix $\mathbf{A}^{-1}$, as shown in.[2] To know the **B** matrix, a cross correlation between the inverse transformed image and an artificially



(a) Rotated image

(b) Estimated WM Autocorrelation

Figure 6. Effect of rotation in the autocorrelation. In (a) there is the rotated image and in (b) the autocorrelation of the estimated watermark of the image in (a). In this example the image was watermarked with multi-tile scheme. The absence of some peaks in relation to Fig. 1 is due to the threshold value.

generated signature will have to be performed.

As it is described in Sec. 3.1 the use of tiling turns into the appearance of a collection of new peaks in the autocorrelation of the estimated watermark (see Fig. 6). The distribution of these peaks corresponds to a very well defined scheme. This scheme can be used to overcome distortions occasionally inferred to the watermark due to attacks. This distortions degrade the peaks pattern leading in some situations to a failure on the detection. Thanks to the presence of additional peaks, the watermarking scheme increases its robustness. Using the relative distances among them, it is possible to check the validity of those, detect the appropriate distribution case (I, II, III or IV) and apply the correct parameters to the inverse transformation.

In this paper only the case where the image has been signed with one key is studied. If several keys were used, another retrieval algorithm for the geometric transformation matrix should be found. In the case where only one key is used, there are mainly four possible cases of peak detection:

Case I. In this case the first peak corresponds to the tile size (this is the peak corresponding to period peak of the autocorrelation), and the second one is the following tile peak (the one corresponding to the second period peak). So the relation between the distances of the first peak and the second will be:

$$\frac{d_1}{d_2} = \frac{tile\_size}{2 \times tile\_size} = 0.5. \tag{27}$$

Case II. The most powerful peak (apart from the central one) is the secondary peak of the central autocorrelation peak (this peak is due to the self-similarity of the watermarking pattern[4,2]) and the second is a tile peak (corresponding to the autocorrelation period described in section 3.1). This case will occur when there is not a large number of tiles in the image (specially in the case of cropping). The relation between the peak distance will, in this case, be:

$$\frac{d_1}{d_2} = \frac{PSHIFT}{tile\_size} = \frac{88}{256} = 0.34375, \tag{28}$$

where *PSHIFT*, which in our case is 88, is the self-similarity distance used in the tests between sub-signatures in the same tile.[2]

Case III. The first peak is a tile peak and the second is a sub-peak around the central one. In this case the relation between distances will be:

$$\frac{d_1}{d_2} = \frac{tile\_size}{PSHIFT} = \frac{256}{88} = 2.90909. \tag{29}$$

Case IV. This is the most complicated case: when only the first peak of the axis is a real one, and the other corresponds to a noisy peak which must not be taken in account. This noisy peak will normally be an interference peak caused by the central one. In this case the relation between peak distances will not be any of the previous values nor an integer multiple of them:

$$\frac{PSHIFT}{noisy\_peak} = \varepsilon \tag{30}$$

with ε any value different from those discussed above.

When this case occurs there is no knowledge about which peak it may be. Two possibilities have to be taken into account:

- The peak found is a peak corresponding to the sub-tile ^{4,2} (the peak is caused by the self-similarity of the watermark in the tile) and so the original distance from the origin is 88 pixels.
- The peak found corresponds to autocorrelation periodization due to tiling (and so the distance of the peak to the origin should be 256 pixels).

Any other combination of peaks is considered as non probable under the effect of geometric attacks, due to the characteristics of the embedding algorithm.

The four cases mentioned above are only for one axis. To extend it to two axis, a combination of them in pairs have to be used. This lends to the 16 cases.

Knowing the relative distance between the detected peaks, we can guess if the most powerful peak (the central peak is not taken into account) is one corresponding to the period of the autocorrelation or to the self-similarity periodicity inside a tile. From that, the adequate $predefined\_distance\_peak$ that should have been found if no attack had been suffered will be applied to **A** in each axis. In Eq. (25), numerators will be the coordinates of the autocorrelation peak found. If no transformation has been applied to the image, the attack matrix should be an identity matrix.

5. CONCLUSIONS

This paper provided evidence that parallelizing a watermarking algorithm offers more advantages than drawbacks. First of all, it optimizes memory and CPU consumption. Second, it also improves watermark robustness, especially to geometric attacks.

The fact of parallelizing the watermarking introduces also some weaknesses, as the possibility to perform the Wiener filter attack. Such weakness can be removed by signing the same image using more than one key. This, though, influences the computation of the attack inversion matrix, since a new peak retrieval algorithm would be necessary.

6. ACKNOWLEDGEMENTS

The authors would like to acknowledge the contributions of Mattia Bertschi, Alexandre Fotinos and Genevieve Dardier for implementing portions of the algorithms described here and for fruitful discussions. Many thanks as well to Pierre Vandergheynst for his comments and suggestions.

REFERENCES

1. E. Debes, G. Dardier, T. Ebrahimi, and A. Herrigel, "Watermarking scheme for large images using parallel processing," *In proceedings of SPIE and IS&T conference on Security and Watermarking of Multimedia Contents III, January 21-26 2001, San Jose, CA* , january 2001.
2. M. Kutter, *Digital Image Watermarking: Hiding Information in Images*. PhD thesis, Signal Processing Laboratory, Swiss Federal Institute of Technology, 1999.
3. M. Kutter, F. Jordan, and F. Bossen, "Digital signature of color images using amplitude modulation," *Journal of Electronic Imaging* **vol. 7**, pp. pp. 326–332, April 1998.
4. M. Kutter, "Watermarking resisting to translation, rotation and scaling," *Proc. of SPIE: Multimedia systems and applications, Boston, USA* **3528**, pp. 423–431, November 1998.
5. J. K. . Ruanaidh and T. Pun, "Rotation, scale and translation invariant spread spectrum digital image watermarking," *Signal Processing* **vol. 66**, pp. 303–317, May 1998.

6. E. Debes, *Exploitation of Parallelism in General Purpose Processor based Systems for Multimedia Applications*. PhD thesis, Signal Processing Laboratory, Swiss Federal Institute of Technology, 2000.
7. J. M. Hart, *Win32 System Programming, Chapter 10*, Addison-Wesley Advanced Windows Series, 1997.
8. J. S. Lim, *Two-Dimensional Signal and Image Processing*, Prentice Hall, 1990.
9. S. Voloshynovskiy, A. Herrigel, N. Baumgrtner, and T. Pun, "A stochastic approach to content adaptive digital image watermarking," *In International Workshop on Information Hiding, Lecture Notes in Computer Science* **vol. 1768**, pp. 212–236, October 1999.
10. A.S.Tanembaum, *Computer Networks*, Prentice-Hall, 1996.
11. J. A. McLaughlin and J. Raviv, "Nth-order autocorrelations in pattern recognition," *Information and Control* **12**(2), pp. 121–142, 1968.
12. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company, September 1993.