Exploiting Self-Similarities to Defeat Digital Watermarking Systems

A Case Study on Still Images

Gwenaël Doërr
doerr@eurecom.fr

Jean-Luc Dugelay
dugelay@eurecom.fr

Lucas Grangé
grange@eurecom.fr

Eurécom Institute
Department of Multimedia Communications
2229, route des Crêtes BP 193
06904 Sophia-Antipolis Cédex
FRANCE

ABSTRACT

Unauthorized digital copying is a major concern for multimedia content providers. Since copyright owners lose control over content distribution as soon as data is decrypted or unscrambled, digital watermarking has been introduced as a complementary protection technology. In an effort to anticipate hostile behaviors of adversaries, the research community is constantly introducing novel attacks to benchmark watermarking systems. In this paper, a generic block replacement attack will be presented. The underlying assumption is that multimedia content is highly repetitive. It should consequently be possible to exploit the self-similarities of the signal to replace each signal block with another perceptually similar one. Alternative methods to compute such a valid replacement block will be surveyed in this paper. Then, experimental results on still images will be presented to demonstrate the efficiency of the presented attack in comparison with other reference image processing operations. Finally, a discussion will be conducted to exhibit the properties that a watermark should have to resist to this attack.

Categories and Subject Descriptors

K.4.1 [**Computer and Society**]: Public Policy Issues—*Intellectual Property Rights*; I.4.9 [**Image Processing and Computer Vision**]: Applications—*Digital Watermarking*; K.6.2 [**Management of Computing and Information Systems**]: Installation Management—*Benchmarks*

General Terms

Security, Reliability

MM-SEC'04 September 20-21, 2004, Magdeburg, Germany.

Keywords

Block replacement attack, self-similarities, intra-signal collusion

1. INTRODUCTION

The recent entrance in the digital world has triggered an increase of multimedia content piracy. Using peer-to-peer networks, it is indeed somewhat easy today to find on the Internet high valued multimedia items (billboard songs, recent movies), download them and copy them on whatever storage device. This drastic lost of royalties has put the multimedia community under pressure to rethink their whole distribution framework. First of all, content providers revised and consolidated their own security policies since nearly 80% of the movie samples available on file sharing networks appeared to have been leaked by industry insiders [1]. On the other hand, initiatives [10, 34] have been launched to deploy and standardize a Digital Rights Management (DRM) technology to protect playback, storage and distribution of multimedia items. The challenge is that encryption alone is not enough to ensure copyright at the client side. As soon as data is decrypted or unscrambled, the adversary obtains a plain-text copy of the multimedia item and can either copy it in its digital form or digitized it from an analog output using an A/D converter. This has motivated the introduction of digital watermarks [6] in almost all modern copyright protection mechanisms.

Digital watermarking basically consists in hiding a key dependent secret signal into digital multimedia data in a robust and imperceptible manner. The robustness of the watermark can be seen as the ability of the detector to retrieve the hidden secret watermark once watermarked data has been altered. For example, the embedded signal should survive D/A-A/D conversion. Watermarks can also be regarded assome signal transmitted along a communication channel (the host multimedia item) whose capacity is exploited to convey more or less information bits. Those parameters (capacity, imperceptibility, robustness) are conflicting and a trade-off has to be found depending on the targeted applications. Introducing watermarks in digital data can be really useful to safeguard copyright. In a *content screening* scenario, content providers insert a secret water-

mark in their multimedia items before releasing them on a public communication channel. On the client side, the media player blindly checks whether the watermark is present or not. In case the secret mark is detected, the player verifies whether it has an authentic and valid license to (dis)play the content. Alternatively, user-specific watermarks denoted as *fingerprints* can be embedded in the data to be protected before being delivered to the customer. Search robots are then deployed to find content copies on the Internet and forensic tools are exploited to identify malicious customers who have broken their license agreement.

In such applications, the embedded watermark either limits the possible usages of multimedia content via playback or copy control, or gives clues on the customer identity which might be potentially used in court. As a result, users are likely to be willing to attack this disturbing technology. Following the old Latin precept *si vis pacem, para bellum*, the research community has made some efforts to anticipate such behaviors. This has resulted in a collection of benchmarking tools [2, 3, 26, 27, 35, 39] which can be used to evaluate the robustness of watermarking techniques. After a short review of the possible attacks on watermarking systems, the generic framework of the block replacement attack is introduced in Section 2. The basic idea consists in replacing each signal block with a perceptually similar one. Several approaches are then investigated in Section 3 to compute a candidate block which can be used for replacement. Finally, experimental results on still image are presented in Section 4 and conclusions are drawn in Section 5 to identify which properties a watermark should have to resist the presented attack.

2. ATTACKS AGAINST DIGITAL WATERMARKING SYSTEMS

There exists a relatively complex tradeoff between conflicting parameters in digital watermarking. As a result, several benchmarks have been released to allow a fair comparison between different algorithms. In particular, efficient attacks have been proposed in an attempt to anticipate malicious behaviors. In content protection systems, embedded watermarks do not add any value to the multimedia items from the customer perspective. On the contrary, the hidden information may be used in court to convince the jury that the sued customer has not respected the license agreement. Therefore, malicious users want to remove this hidden piece of evidence and design efficient attacks to defeat the system. This is very similar to the situation in cryptography: major advances come from the competition between hackers who try to beat down the security system and system designers who create new countermeasures to survive to new attacks. Attacks against digital watermarking systems are consequently reviewed in the next subsections. In particular, two major classes of attacks are isolated: watermark removal attacks (Subsection 2.1) and synchronization removal attacks (Subsection 2.2). There also exists some cryptographic and protocol attacks [38] but they will not be considered here since they are beyond the scope of this paper. Then, a new alternative attack is presented in Subsection 2.3 and is further investigated in the remainder of the paper.

2.1 Watermark Removal Attacks

In digital watermarking, the detector usually computes a score, e.g. a correlation score. It is then compared to a threshold to assert whether a watermark is present or not. A potential target of an adversary is consequently to find an attack which brings the detection score below the detection threshold so that the embedded watermark is no longer detected. In other terms, the attack aims at decreasing the power of the hidden watermark down to a level where the detector cannot *reliably* assert that it is present. A large range of signal processing operations can be considered as removal attacks. Low-pass filtering and lossy compression are likely for instance to alter watermarks since they are usually located in high frequencies. However, many other primitives have to be considered to obtain a fair benchmark [23] including gamma correction, quantization, noise addition due for example to D/A-A/D conversion (printing and scanning)... Beside such blind operations, a new brand of attacks based on the estimation theory has appeared. The basic idea consists in estimating either the original unwatermarked content or the embedded watermark itself. For example, denoising techniques can be exploited to remove a hidden watermark. However, such approaches have been shown to introduce annoying blurring artifacts and a two-steps strategy is usually preferred. The attacker first computes an estimate of the embedded watermark using for example local median filtering [24] or Wiener filtering [36]. This estimation can then be processed, e.g. high-pass filtered [24], to remove unlikely low-frequency components. Finally, the estimated watermark is remodulated either with a constant strength[24, 36] or an adaptive one [37] which considers perceptual constraints. Furthermore, if the attacker has access to several documents carrying the same watermark, successful collusion attacks [9, 15] can be designed to obtain a refined estimation of the embedded watermark, and thus a more efficient attack after remodulation.

2.2 Synchronization Removal Attacks

This second class of attacks does not explicitly aim at removing the embedded watermark. It rather tries to disrupt the communication between the embedder and the detector. To this end, the attacker basically performs operations which desynchronize the detector. Indeed, many detectors today are correlation based and thus expect each watermark sample to be at a predefined location in the working space. This knowledge is shared by both the embedder and the detector. If an external party disturbs this alignment, the convention known by the detector becomes obsolete and communication is no longer possible. In other terms, the detector needs to be synchronized with the embedder to detect the hidden watermark. Consequently, spatial and temporal alterations can be performed on the watermarked data to trap the detector. Examples of such operations include image flipping, rotations, cropping, scaling, time stretching... Countermeasures to such attacks basically exploit either an invariant embedding domain [28], or a known template used for registration i.e. for resynchronization [22]. In case a registration pattern is used, it should be noted that this can introduce some undesired visible peaks in the frequency domain. As a result, an attacker can isolate and erase those peaks to remove the resynchronization signal [14]. Furthermore, if robustness to global transformation is almost solved nowadays, local random geometric distortions are more com-

Figure 1: Stirmark attack (left: original, right: attacked): the original image is submitted to random local geometrical distortions. However, the resulting image has not lost its commercial value.

plicated to address. The most well-known implementation of such an attack is the random bending attack [29]. It basically exploits the fact that the human visual system is not sensitive against shifts and local affine transformations. Therefore, pixels can be locally shifted, scaled and rotated without significant visual distortions. The impact of this attack is depicted in Figure 1. When comparing the underlying grid, it is somewhat obvious that there is a difference between the two images but this statement is far less straightforward if only the central part is considered. In other terms, this attack *does not remove the commercial value of the picture*. However, it still succeeds in trapping most of the watermark detectors today.

2.3 Block Replacement Attacks

Both classes of attacks previously presented exhibit some shortcomings. On one side, watermark estimation remodulation attacks basically rely on the assumption that it is possible to obtain a somewhat *good* estimation of the embedded watermark. If such a refined estimation can be computed when several watermarked documents are colluded [9, 15], it is relatively difficult to do when a single watermarked document is considered. On the other side, desynchronization attacks do not remove the hidden watermark. It simply alters the alignment shared by the embedder and the detector. Nevertheless, nothing ensures that a future enhanced version will still not be able to detect a desynchronized watermark. The introduction of a misalignment also prevents from using common quantitative metrics such as the Peak Signal to Noise Ratio (PSNR) to evaluate the impact of the attack. For instance, with the random bending attack, the PSNR is likely to be very low even if the watermarked and attacked images are perceptually similar.

Those limitations have consequently motivated the introduction of a novel attack. Ideally, the attack would consist of blindly restoring the original document from the watermarked one. However such a perfect attack is impossible to implement in practice. In this paper, the goal will consequently be to design an attack which has the following specifications:

1. After the attack, the detector is no longer able to detect the embedded watermark.

2. The attack does not introduce any geometric distortion so that quantitative measures of distortion between the watermarked and attacked documents remain pertinent.

3. The attack introduces a fair additional distortion. The distance between the watermarked and attacked documents should be close to the distance existing between the original and watermarked documents.

4. The attack is designed in such a way that it is possible to adapt the strength of the attack. Indeed, alternative watermarking schemes insert their watermark with a different embedding strength. As a result, it is necessary to tune the strength of the attack according to the embedding strength.

5. The attack ensures that a future improved version of the detector alone cannot overcome the problem. The protection of the watermarked documents is definitely lost and technology providers have to rework both embedder and retriever.

Multimedia digital data is highly redundant: successive video frames are highly similar in a movie clip, most songs contain today some repetitive patterns. An attacker can consequently exploit those similarities to successively replace each part of the signal with a *similar* one taken from another location in the same signal. Such approaches have already been investigated to obtain efficient compression tools [11]. The remainder of this article is consequently devoted to the description of possible implementations of such block replacement attacks whose generic framework is given in Table 1. The signal to be processed is first partitioned into a set of blocks $\mathbf{B}_T$ of size S_T. Those blocks can either overlap or not. The asset of using overlapping blocks is basically that it prevents strong blocking artifacts on the border of the blocks by averaging the overlapping areas. The attack process then each one of those blocks sequentially.

Table 1: Generic Description of the Block Replacement Attack

1	Partition the signal in blocks $\mathbf{B}_T$ of size S_T
2	For each block,
	(a) Define a search window and build a codebook $\mathbf{Q}$ which contains a set of blocks $\mathbf{B}_{Q_i}$ of size S_Q
	(b) Compute a replacement block $\mathbf{B}_R$ *similar* to $\mathbf{B}_T$ using the blocks in $\mathbf{Q}$
	(c) Replace $\mathbf{B}_T$ by $\mathbf{B}_R$

For each block, a search window is defined. It can be chosen in the vicinity of the block $\mathbf{B}_T$ or randomly for security reasons. This search window is then partitioned to obtain a codebook $\mathbf{Q}$ of blocks $\mathbf{B}_{Q_i}$ of size S_Q. Once again, those blocks can overlap or not. Furthermore, following previous work on fractal coding [16], the size S_Q can be different from the size S_T of the original blocks. In this case, block resizing is necessary. Additionally, the codebook can also be artificially enlarged by introducing geometrically transformed versions (identity, 4 flips, 3 rotations) of the blocks in the search window. Indeed, the larger the codebook $\mathbf{Q}$ is, the more choices there will be to compute a replacement block $\mathbf{B}_R$ which is *similar* to the block $\mathbf{B}_T$ to be replaced. On the other hand, the larger the codebook $\mathbf{Q}$ is, the higher

the computational complexity is and a trade-off has to be found. In this paper, the Mean Square Error (MSE) will be used to evaluate how similar are two blocks. It is computed as follows:

$$\mathrm{MSE}(\mathbf{B}_R, \mathbf{B}_T) = \frac{1}{S_T} \sum_{i=1}^{S_T} \left(\mathbf{B}_R(i) - \mathbf{B}_T(i)\right)^2 \qquad (1)$$

where the index i in the summation can be one-dimensional (sound) or multidimensional (image, video). The lower the MSE is, the more similar are the two blocks. Finally, the original block $\mathbf{B}_T$ is substituted by the computed replacement block $\mathbf{B}_R$.

3. COMPUTATION OF THE CANDIDATE BLOCK FOR REPLACEMENT

Once the codebook $\mathbf{Q}$ has been built, the next step of the attack consists in producing a candidate replacement block $\mathbf{B}_R$ which is similar to the target block $\mathbf{B}_T$ using the blocks $\mathbf{B}_{Q_i}$ of the codebook. To this end, several approaches can be investigated. In Subsection 3.1, error concealment techniques are introduced as a possible solution to achieve this goal. Next, a desynchronization strategy is presented in Subsection 3.2 which basically aims at shuffling the watermark samples while keeping the host data synchronized. Alternatively, in Subsection 3.3, blocks from the codebook are optimally colluded to produce a valid replacement block. Finally, Subsection 3.4 presents an approach which exploits space dimension reduction techniques.

3.1 Block Restoration

Error concealment techniques have initially been designed to recover blocks which have been lost or corrupted during digital transmission. As depicted in Figure 2, when a missing block is detected, the neighborhood of this block is considered to obtain a prediction of the missing information. Such approaches can be exploited to design an efficient block replacement attack. Sequentially, each block of the signal is considered as missing and the error concealment procedure computes a replacement block [40]. However, this strategy suffers from two major shortcomings. First, there is no direct attacking strength i.e. there is no possibility to adapt the impact of the attack according to the watermarking strength. Second, each block is considered as *missing* which is not really the case. In other terms, some information is ignored and it is likely to result in a relatively poor quality attacked signal. For both those reasons, such approaches will not be further considered in the remainder of this paper.

3.2 Block Swapping

Most watermarking algorithms have exhibited weaknesses against desynchronization attacks and especially non global ones. The random bending attack [29] has been considered for a long time now as a reference for benchmarking watermarking systems. However, countermeasures have appeared which basically exploit the fact that this processing does not drastically modify the *geography* of the embedded watermark. Each watermark sample is slightly displaced but it remains in the neighborhood of its original location. As a result, local block-matching based detectors [13, 19, 31] have been shown to be able to recover watermarks altered by such attacks. Alternatively, the block swapping attack [32] aims

Figure 2: Error concealment techniques: when a block is detected as corrupted or missing, it is discarded and the algorithm tries to predict it using blocks in the vicinity.

at shuffling the watermark samples while keeping the host data synchronized. The basic idea is to replace each block of the signal with a similar one, which *does not carry the same watermark signal*. In other terms, the *geography* of the embedded watermark is strongly altered so that resynchronization is no longer possible, and thus the detector is confused.

Table 2: Block Swapping Attack

	For each block $\mathbf{B}_T$ of the signal
1	Build the block codebook $\mathbf{Q}$
2	Perform photometric compensation
3	Sort the blocks $\mathbf{B}_{Q_i}$ according to the MSE
4	Set $\mathbf{B}_R$ as the most similar block
5	Replace $\mathbf{B}_T$ by $\mathbf{B}_R$

The pseudo-code of the block swapping attack is detailed in Table 2. For each block $\mathbf{B}_T$ of the input signal, a search window is defined and a codebook $\mathbf{Q}$ built. Next, photometric compensation is necessary, at least with still images, to obtain a good pool of candidate blocks for replacement. Otherwise, the codebook $\mathbf{Q}$ is unlikely to contain a block which is similar enough to $\mathbf{B}_T$ and the replacement process will introduce a strong distortion. As a result, each block $\mathbf{B}_{Q_i}$ is transformed in $s\mathbf{B}_{Q_i} + o\mathbf{1}$, where $\mathbf{1}$ is a block containing only ones, so that the MSE with the target block $\mathbf{B}_T$ is minimized. This is a simple least squares problem and the scale s and offset o can be determined as follows:

$$s = \frac{(\mathbf{B}_T - \mathrm{m}_T\mathbf{1}) \cdot (\mathbf{B}_{Q_i} - \mathrm{m}_{Q_i}\mathbf{1})}{|\mathbf{B}_{Q_i} - \mathrm{m}_{Q_i}\mathbf{1}|^2} \qquad (2)$$

$$o = \mathrm{m}_T - s.\mathrm{m}_{Q_i} \qquad (3)$$

where m_T (resp. m_{Q_i}) is the mean value of block $\mathbf{B}_T$ (resp. $\mathbf{B}_{Q_i}$), $\cdot$ is the linear correlation defined as:

$$\mathbf{B} \cdot \mathbf{B}' = \frac{1}{S_T} \sum_{i=1}^{S_T} \mathbf{B}(i)\mathbf{B}'(i) \qquad (4)$$

and $|\mathbf{B}|$ is the norm defined as $\sqrt{\mathbf{B} \cdot \mathbf{B}}$. At this point, the transformed blocks $s\mathbf{B}_{Q_i} + o\mathbf{1}$ are sorted in ascending order according to their similarity with the target block $\mathbf{B}_T$. The most similar block is then retained and used for replacement. In this version, the block replacement attack is equivalent

to image compression with a fractal coder [11]. A visual interpretation of this attack is depicted in Figure 3. In the same fashion, an alternative approach consists in building iteratively sets of similar blocks and randomly shuffling their positions [30, 20] until all the blocks have been replaced.

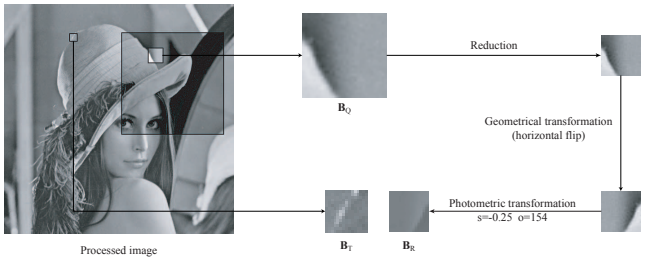

Figure 3: Block swapping attack: each block is replaced by the one in the search window which is the most similar modulo a geometrical and photometric transformation.

Performing photometric compensation and computing the MSE can become computationally prohibitive as the number of blocks in the codebook $\mathbf{Q}$ increases. Furthermore, there is no real need to perform explicitly photometric compensation for each block $\mathbf{B}_{Q_i}$. In fact, photometric compensation needs to be done only for a single block of the codebook, the one which will be used for replacement. There exists a relationship between $\mathrm{MSE}(s\mathbf{B}_{Q_i} + o\mathbf{1}, \mathbf{B}_T)$ and the correlation coefficient:

$$\mathbf{B}_{Q_i} \odot \mathbf{B}_T = \frac{\mathbf{B}_{Q_i} - \mathrm{m}_{Q_i}\mathbf{1}}{|\mathbf{B}_{Q_i} - \mathrm{m}_{Q_i}\mathbf{1}|} \cdot \frac{\mathbf{B}_T - \mathrm{m}_T\mathbf{1}}{|\mathbf{B}_T - \mathrm{m}_T\mathbf{1}|} \qquad (5)$$

After a few derivations, the following equation can be obtained:

$$\mathrm{MSE}(s\mathbf{B}_{Q_i}+o\mathbf{1}, \mathbf{B}_T) = |\mathbf{B}_T-\mathrm{m}_T\mathbf{1}|^2\Big(1-(\mathbf{B}_{Q_i}\odot\mathbf{B}_T)^2\Big) \quad (6)$$

It means that sorting the blocks in ascending $\mathrm{MSE}(s\mathbf{B}_{Q_i} + o\mathbf{1}, \mathbf{B}_T)$ is equivalent to sorting the blocks in descending $(\mathbf{B}_{Q_i} \odot \mathbf{B}_T)^2$. This property can be exploited to sort the blocks of the codebook without explicitly building the modified blocks $s\mathbf{B}_{Q_i} + o\mathbf{1}$.

Exchanging highly similar blocks is likely to be imperceptible. However, it is also likely not to affect the watermark signal. A threshold τ_{low} can consequently be introduced to force a minimum distortion between the replacement block $\mathbf{B}_R$ and the target block $\mathbf{B}_T$ which is to be replaced. In other terms, the step 3 is modified so that the replacement block is no longer the most similar block in the codebook $\mathbf{Q}$ modulo a geometrical and photometric transformation, but rather the most similar block *whose distortion is if possible above* τ_{low}. This additional parameter can be regarded as an attacking strength and introduces a trade-off between the efficiency of the attack and its impact in terms of fidelity.

3.3 Blocks Combination

In the previous subsection, a threshold τ_{low} has been introduced to ensure that the replacement block $\mathbf{B}_R$ is not similar to the target block $\mathbf{B}_T$ up to the point that it also contains the watermark signal. On the other hand, there is no guaranty that this replacement block will be similar enough to be imperceptible once it has been substituted with the original block. In fact, experimental results have shown that blocks are likely to be badly estimated with a single block, even if photometric compensation is performed. Following previous advances in fractal coding[12, 25], the idea is then to combine several blocks $\mathbf{B}_{Q_i}$ in the codebook $\mathbf{Q}$ to obtain a better replacement block:

$$\mathbf{B}_R = \sum_{i=1}^{N} \lambda_i \mathbf{B}_{Q_i} \qquad (7)$$

where the λ_i's are mixing coefficients. To obtain the best possible replacement block, those mixing coefficients are chosen so that $\mathrm{MSE}(\mathbf{B}_R, \mathbf{B}_T)$ is minimized[1]. This is a traditional least squares problem which can be easily solved using common linear algebra tools. From this novel perspective, the block replacement attack is more related with intra-signal collusion attacks [9] i.e. combining several watermarked contents to obtain unwatermarked content.

3.3.1 *Fixed number of blocks:*

With this new approach in mind, a novel block replacement attack can be designed as depicted by the pseudo-code given in Table 3. For each block $\mathbf{B}_T$ of the input signal, a search window is defined and a codebook $\mathbf{Q}$ built. Then the blocks $\mathbf{B}_{Q_i}$ are sorted in ascending order according to their similarity with the target block $\mathbf{B}_T$ using Equation (6). At this point, a *fixed* number of blocks e.g. the first N blocks in the codebook $\mathbf{Q}$ are considered to compute an optimal replacement block $\mathbf{B}_R$ in a least squares sense [21]. Finally, this candidate block is put into the place of the original one $\mathbf{B}_T$. Here again, the step 3 can be modified to prevent the candidate replacement block $\mathbf{B}_R$ from being too similar to the target block $\mathbf{B}_T$. To this end, the first N blocks *whose distortion is above a threshold* τ_{low} can be considered, rather than the N first ones, to compute the optimal replacement block. The expectation is that using poorer blocks from the codebook $\mathbf{Q}$ will output a poorer candidate block for replacement.

Table 3: Fixed Number of Blocks Combination Attack

For each block $\mathbf{B}_T$ of the signal	
1	Build the block codebook $\mathbf{Q}$
2	Sort the blocks $\mathbf{B}_{Q_i}$
3	Build the optimal replacement block $\mathbf{B}_R$ using the first N blocks in $\mathbf{Q}$
4	Replace $\mathbf{B}_T$ by $\mathbf{B}_R$

However, the attacker would rather like to be able to ensure that the final distortion $\mathrm{MSE}(\mathbf{B}_R, \mathbf{B}_T)$ is between the values τ_{low} and τ_{high}. Indeed, the replacement block should not be *too good* ($\mathrm{MSE}(\mathbf{B}_R, \mathbf{B}_T) < \tau_{\text{low}}$). Otherwise, it is also likely to carry the watermark signal. Furthermore, it should not be *too bad* either ($\mathrm{MSE}(\mathbf{B}_R, \mathbf{B}_T) > \tau_{\text{high}}$) so that the block replacement attack does not introduce perceptible artifacts. Unfortunately, it is difficult to predict this distortion from the distortions of the blocks $\mathbf{B}_{Q_i}$ used for combination. It can only be checked a posteriori. Figure 4 shows

[1]It should be noted that the block $\mathbf{1}$ can be artificially added to the codebook $\mathbf{Q}$ to permit automatic mean value adjustment.

$N = 1$ $N = 10$ $N = 30$ $N = 60$

Figure 4: Influence of the number of blocks N used for combination once the thresholds τ_{low} and τ_{high} have been set. Light (resp. dark) gray blocks indicate *too good* (resp. *too bad*) blocks.

the localization of the *too good* and *too bad* blocks once the two thresholds τ_{low} and τ_{high} have been fixed and that the number N of blocks used for combination is varying. The first observation is that the number of *too bad* blocks decreases as N increases, while the number of *too good* blocks increases. Secondly, the number of blocks needed to make the distortion $\text{MSE}(\mathbf{B}_R, \mathbf{B}_T)$ drop below τ_{high} seems to be related with the content of blocks: flat blocks require fewer blocks to obtain a valid replacement block $\mathbf{B}_R$ after combination in comparison with textured blocks. This calls for a new approach which automatically adjusts the number of blocks used for combination.

3.3.2 Adaptive number of blocks:

The previous subsection has highlighted the fact that using a *fixed* number of blocks is somewhat limiting. Each block does not indeed need the same number of blocks to be finely enough approximated e.g. flat vs. textured blocks. An improved algorithm whose pseudo-code is given in Table 4 is consequently introduced so that the number and the set of blocks chosen for combination are adaptively modified to obtain a candidate replacement block $\mathbf{B}_R$ whose distortion $\text{MSE}(\mathbf{B}_R, \mathbf{B}_T)$ is between τ_{low} and τ_{high}. The basic idea is to modify the step 3 in the previous algorithm by checking the distortion $\Delta = \text{MSE}(\mathbf{B}_R, \mathbf{B}_T)$ of the computed candidate block for replacement. Depending on the value of this distortion, different rules are enforced.

- If Δ is between τ_{low} and τ_{high}, a valid candidate block has been found for replacement. A flag is consequently set to 1 to terminate the adaptive algorithm.

- If Δ is greater than τ_{high}, it means that the obtained candidate block for replacement does not approximate the target block $\mathbf{B}_T$ well enough. As a result, the attack would introduce perceptible artifacts if they were substituted. N is consequently incremented so that more blocks are considered during combination and thus a better candidate block is obtained.

- If Δ is lower than τ_{low}, the candidate replacement block $\mathbf{B}_R$ is too similar to the target block $\mathbf{B}_T$. It is likely to also carry the watermark signal. The offset Φ is consequently incremented so that poorer blocks from $\mathbf{Q}$ are considered during block combination. Furthermore, the number of combined blocks N is reset to 1.

It should be noted that this algorithm inherently assumes that a candidate block whose distortion falls within the bounds τ_{low} and τ_{high} will be found. However, nothing ensures that it will be the case in practice. In particular, for small codebooks or close threshold values, such a block might not exist. The algorithm consequently needs to be slightly modified to handle such exceptions. For example, if this case occurs, the candidate block whose distortion minimizes $\max(\sqrt{\tau_{\text{low}}} - \sqrt{\Delta}, \sqrt{\Delta} - \sqrt{\tau_{\text{high}}})$ can be retained for replacement.

Table 4: Adaptive Number of Blocks Combination Attack

For each block $\mathbf{B}_T$ of the signal

1. Build the block codebook $\mathbf{Q}$
2. Sort the blocks $\mathbf{B}_{Q_i}$
   Set $\Phi = 0$, $N = 1$, flag $= 0$
3. While (flag $= 0$) AND ($\Phi + N \leq |\mathbf{Q}|$)
   (a) Build the optimal replacement block $\mathbf{B}_R$ using N successive blocks from $\mathbf{Q}$ starting with block $\mathbf{B}_{Q_{\Phi+1}}$
   (b) Compute $\Delta = \text{MSE}(\mathbf{B}_R, \mathbf{B}_T)$
   (c) If $\tau_{\text{low}} \leq \Delta \leq \tau_{\text{high}}$, set flag $= 1$
   (d) Else if $\Delta > \tau_{\text{high}}$, increment N
   (e) Else increment Φ and reset N to 1
4. Replace $\mathbf{B}_T$ by $\mathbf{B}_R$

3.4 Block Projection

The previous attack gives some good results as will be reported in Section 4. However, it is in some sense suboptimal. In step 3, when the computed candidate block for replacement is found to be too similar to the target block $\mathbf{B}_T$, the offset Φ is incremented to consider poorer blocks during combination. Nevertheless, this does not ensure that a poorer block will be obtained *after combination*. In fact, this is only a way of getting alternative candidate blocks for replacement until one is found to be in the target interval $[\tau_{\text{low}}, \tau_{\text{high}}]$. In this case, all the possible blocks combinations should be computed instead of a restricted subset. But this is not possible in practice because of the prohibitive computational cost. As a result, a substitute approach is investigated below.

Finding the mixing coefficients λ_i which minimizes the distortion $\text{MSE}(\sum_{i=1}^{N} \lambda_i \mathbf{B}_{Q_{\Phi+i}}, \mathbf{B}_T)$ is equivalent to com-

puting the coordinates of the target block $\mathbf{B}_T$ in the subspace spanned by the N blocks $\mathbf{B}_{Q_{\Phi+i}}$. In other terms, the block replacement attack comes down to finding a subspace $\mathcal{S}$ for each block $\mathbf{B}_T$ so that $\text{MSE}(\mathbf{B}_T^{\mathcal{S}}, \mathbf{B}_T)$ is between τ_{low} and τ_{high}, where $\mathbf{B}_T^{\mathcal{S}}$ is the projection of the block $\mathbf{B}_T$ onto the subspace $\mathcal{S}$. In the approaches described in Subsection 3.3, most of the computational cost is due to the fact that the basis vectors of the subspace $\mathcal{S}$ - in this case the blocks $\mathbf{B}_{Q_i}$ of the codebook $\mathbf{Q}$ - are not orthogonal. Thus, a least squares problem has to be solved to obtain the coordinates λ_i's of the target block in the generated subspace $\mathcal{S} = \text{span}\{\mathbf{B}_{Q_i}\}$. The problem would have been much easier if the basis vectors were orthogonal: successive projections on each vector gives then the coordinates. This has consequently motivated further research to investigate how to obtain such orthogonal basis. In particular, approaches exploiting Gram-Schidt Orthonormalization (GSO) and Principal Component Analysis (PCA) have been surveyed.

3.4.1 Gram-Schmidt Orthonormalization

The Gram-Schmidt orthonormalization procedure takes a non-orthogonal set of linearly independent vectors and constructs an orthogonal basis [4]. Thus, the goal is to incorporate it into a framework which iteratively builds an orthogonal basis in a *best possible match* fashion. First, the algorithm finds the block $\mathbf{B}_{Q_i}$ in $\mathbf{Q}$ which minimizes:

$$\text{MSE}(\mathbf{B}_T, \lambda_i \mathbf{B}_{Q_i}) \quad \text{with} \quad \lambda_i = \frac{\mathbf{B}_T \cdot \mathbf{B}_{Q_i}}{|\mathbf{B}_{Q_i}|^2} \tag{8}$$

Once this optimal block has been found, it is inserted into the basis $\{S_i\}$ which spans the subspace $\mathcal{S} = \text{span}\{\mathbf{S}_i\}$. Next, both the target block $\mathbf{B}_T$ and the codebook $\mathbf{Q}$ are projected onto the subspace orthogonal to $\mathcal{S}$ as follows:

$$\mathbf{B}^{\mathcal{S}^\perp} = \mathbf{B} - \sum_{\mathbf{S}_i \in \mathcal{S}} \frac{\mathbf{B} \cdot \mathbf{S}_i}{|\mathbf{S}_i|^2} \mathbf{S}_i \tag{9}$$

where $\mathbf{B}$ is some original input block and $\mathbf{B}^{\mathcal{S}^\perp}$ its projection on $\mathcal{S}^\perp$. Then, the algorithm search for the best block as in Equation (8) and it is inserted into the basis which spans the subspace $\mathcal{S}$. The algorithm iterates until the distortion $\text{MSE}(\mathbf{B}_T, \mathbf{B}_T^{\mathcal{S}})$ between the target block $\mathbf{B}_T$ and its projection on the constructed subspace $\mathcal{S}$ falls within the interval $[\tau_{\text{low}}, \tau_{\text{high}}]$. Nevertheless, this approach has two major shortcomings. First, the whole procedure requires many projection and correlation computations, which is likely to rapidly become intractable as the size of the codebook grows. Second, the basis is iteratively built in a *best possible match* way. However, nothing ensures that combining two blocks, which have been successively found to be the best possible match, will output a better candidate block than another combination of two blocks in the codebook.

3.4.2 Principal Component Analysis

Principal Component Analysis [17] basically takes a set of vectors and outputs its centroid and a set of eigenvectors which can be seen as the directions of variations of the vectors in the set. Furthermore, each eigenvector is associated with an eigenvalue which indicates how much the set of vectors varies in this direction. The higher the eigenvalue, the more variations in the associated direction. Such a procedure can be exploited to design an efficient block replacement attack as depicted in Table 5. First, a PCA is performed considering the different blocks $\mathbf{B}_{Q_i}$ in the codebook $\mathbf{Q}$. This gives a centroid $\mathbf{C}$ defined as follows:

$$\mathbf{C} = \frac{1}{|\mathbf{Q}|} \sum_{\mathbf{B}_{Q_i} \in \mathbf{Q}} \mathbf{B}_{Q_i} \tag{10}$$

and a set of eigenblocks $\mathbf{E}_i$ associated with their eigenvalues e_i. Those eigenblocks are then sorted by descending eigenvalues i.e. the direction $\mathbf{E}_1$ contains more information than any other one in the basis. Then, a candidate block for replacement $\mathbf{B}_R$ is computed using the N first eigenblocks so that the distortion with the target block $\mathbf{B}_T$ is minimized. In other terms, the block $\mathbf{B}_T - \mathbf{C}$ is projected onto the subspace spanned by the N first eigenblocks. As a result, the replacement block can be written:

$$\mathbf{B}_R = \mathbf{C} + \sum_{i=1}^{N} \frac{(\mathbf{B}_T - \mathbf{C}) \cdot \mathbf{E}_i}{|\mathbf{E}_i|^2} \mathbf{E}_i \tag{11}$$

Of course, the distortion $\Delta = \text{MSE}(\mathbf{B}_T, \mathbf{B}_R)$ gracefully decreases as the number N of combined eigenblocks increases. Thus, an adaptive framework is introduced to identify which value N should have so that the distortion Δ falls within the range $[\tau_{\text{low}}, \tau_{\text{high}}]$. It may happen that no value of N gives a candidate block within this interval. In this case, the value N which gives the candidate block whose distortion minimizes $\max(\sqrt{\tau_{\text{low}}} - \sqrt{\Delta}, \sqrt{\Delta} - \sqrt{\tau_{\text{high}}})$ is retained. The major interest of this method is that it considers the *whole* codebook $\mathbf{Q}$ to compute the orthogonal basis used for projection. Furthermore, experiments have shown that it was slightly quicker than the attack presented in Subsection 3.3. It should be noted that the underlying assumption is that most of the watermark energy will be concentrated in the last eigenblocks since the watermark can be seen as details. As a result, if a valid candidate block can be built without using the last eigenblocks, the watermark signal will not be reintroduced.

Table 5: Block Projection on a PCA-Defined Subspace Attack

For each block $\mathbf{B}_T$ of the signal

1. Build the block codebook $\mathbf{Q}$
2. Perform photometric compensation
3. Performs the PCA of the blocks in $\mathbf{Q}$ to obtain a set of orthogonal eigenblocks $\mathbf{E}_i$ associated with their eigenvalues e_i
   Set $N = 1$, flag $= 0$
4. While (flag $= 0$) AND ($N \leq S_T$)
   (a) Build the optimal replacement block $\mathbf{B}_R$ using the eigenblocks $\mathbf{E}_i$ associated with the first N eigenvalues
   (b) Compute $\Delta = \text{MSE}(\mathbf{B}_R, \mathbf{B}_T)$
   (c) If $\tau_{\text{low}} \leq \Delta \leq \tau_{\text{high}}$, set flag $= 1$
   (d) Increment N
5. Replace $\mathbf{B}_T$ by $\mathbf{B}_R$

4. EVALUATION OF THE ATTACK

The description of the different block replacement attacks has been kept general on purpose. No hypothesis has been made on the data to be processed to offer a generic framework. This attack can consequently be applied to different

types of multimedia. Previous work from Microsoft has focused on audio data [20, 21, 30]. Temporal Frame Averaging after Registration in video [7, 8] can also be regarded as some sort of block replacement attack which exploits redundancy in successive video frames. In this paper, image documents will be considered as an extension of early work [32]. The next subsections introduce the enforced watermarking scheme during the experiments as well as two basic signal processing operations which will be used as references. Finally, the efficiency of the different proposed algorithms is surveyed in the last subsection.

4.1 Watermarking Scheme

A basic additive spread spectrum watermark has been considered during the experiments. A secret key K is used as a seed to generate a pseudo-random watermark pattern $\mathbf{W}(K)$, whose samples have zero mean and unit variance. This watermark is then scaled by an embedding strength α and added in the spatial domain to the luminance component $\mathbf{I}_{\mathrm{o}}$ of the original image as follows:

$$\mathbf{I}_{\mathrm{w}} = \mathbf{I}_{\mathrm{o}} + \alpha \mathbf{W}(K) \quad \mathbf{W}(K) \sim \mathcal{N}(0,1) \tag{12}$$

where $\mathbf{I}_{\mathrm{w}}$ is the resulting watermarked luminance component. Perceptual shaping can be introduced to improve the invisibility of the watermark by making for example the embedding strength α dependent of the local content of the host image. In practice a global embedding strength equal to 3 has been used which results in a distortion of 9 in terms of MSE, or 38 dB in terms of Peak Signal to Noise Ratio (PSNR).

On the other side, when an image is presented to the detector for verification, the pseudo-random watermark $\mathbf{W}(K)$ is re-generated using the shared secret key K. Then, the detector computes a simple linear correlation as follows:

$$\rho(\mathbf{I}, K) = \mathbf{I} \cdot \mathbf{W}(K) = \big(\mathbf{I}_{\mathrm{o}} + \epsilon\alpha\mathbf{W}(K)\big) \cdot \mathbf{W}(K) \approx \epsilon\alpha \tag{13}$$

where ϵ is equal to 1 or 0 depending if the tested luminance component $\mathbf{I}$ is watermarked or not. If host interference $(\mathbf{I}_{\mathrm{o}} \cdot \mathbf{W}(K))$ is neglected, the correlation score should be equal to α when the watermark $\mathbf{W}(K)$ is present in the tested image, while it should be almost equal to zero if $\mathbf{W}(K)$ has not been embedded. In practice, host interference can be cancelled in a preprocessing step [5] during embedding to enhance the detection statistics. Finally, the correlation score is compared to a threshold τ_{detect} to assert whether or not the watermark $\mathbf{W}(K)$ has been embedded. This threshold can for example be set to $\alpha/2$ to have equal false positive and false negative probabilities.

4.2 Reference Attacks

For comparison, the impact of two reference attacks will also be reported. Since watermarking is done in the luminance component of the images, attacks will also be performed only on the luminance component. First, linear filtering and in particular Gaussian filtering has been considered. The filters are computed as follows:

$$\mathrm{G}_\sigma[x,y] = \frac{\mathrm{g}_\sigma[x,y]}{\sum_{x,y} \mathrm{g}_\sigma[x,y]} \quad \text{with} \quad \mathrm{g}_\sigma[x,y] = e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{14}$$

where σ is the width of the Gaussian filter. The range of x, y is limited so that all large values of $\mathrm{G}_\sigma[x,y]$ are included. The filtered image is then obtained by convolving the image with the computed filter. The larger the filter width is, the more distorted is the filtered image. The second reference attack is lossy compression and especially JPEG compression [18]. This standard specifies the quantization values for DCT coefficients by multiplying a quantization matrix (Table 6) by a global quantization level Q, which is related to a user specified *quality factor* QF in the range of 0 to 100:

$$Q = \begin{cases} 50/QF & \text{if } QF < 50 \\ 2 - 0.02\,QF & \text{if } QF \geq 50 \end{cases} \tag{15}$$

For example, if $QF = 25$, the global quantization level is equal to 2 and the DC term is quantized with a quantization level of $q = 32$. In JPEG, loss of information only occurs during quantization of DCT coefficients. As a result, it is sufficient to perform this quantization to simulate the effects of JPEG compression. The following operation is performed to obtain the quantized value $\bar{x}$ of a DCT coefficient x

$$\bar{x} = q \left\lfloor \frac{x}{q} + 0.5 \right\rfloor \tag{16}$$

where q is the quantization value computed as described above. The lower the JPEG quality factor is, the more distorted is the compressed image.

Table 6: Luminance Quantization Matrix used in JPEG

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
48	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

4.3 Performance

A database of 500 images of size 512×512 has been considered for experiments. It contains snapshots, synthetic images, drawings and cartoons. All the images are first watermarked using the algorithm described in Subsection 4.1. Since the detection is based on the computation of a correlation score, distortion vs. correlation curves can be plotted to study the impact of a given attack. To this end, each watermarked images has been submitted to 4 alternative attacks (Gaussian blurring, DCT quantization, adaptive number of blocks combination, and block projection on PCA-defined subspace) with predefined attacking parameter settings. For the reference attacks, the width σ of the filter and the quality factor QF can be varied. On the other hand, for both block replacement attacks, the thresholds τ_{low} and τ_{high} have to be set. However they can be set equal so that the resulting parameter $\tau_{\mathrm{target}} = \tau_{\mathrm{low}} = \tau_{\mathrm{high}}$ basically sets a target distortion in terms of MSE that the attack should introduce. Furthermore, 8×8 blocks have been used with a 4-pixels overlapping. Using overlapping blocks is indeed really important to avoid annoying blocking artifacts with high values for τ_{target}. At this point, for each image in the database, a distortion vs. correlation curve can be drawn for each one of the 4 surveyed attacks. The different curves associated with a given attack are then *averaged* to obtain a single curve per attack which depicts the statistical behavior

of the image database for a particular attack. The obtained 4 curves are reported in Figure 5.

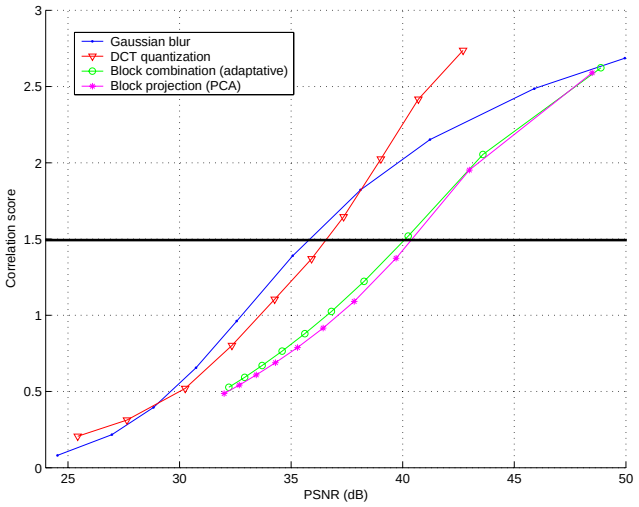

Figure 5: Correlation score vs. distortion curves for the different surveyed attacks.

The goal of the attacker is to decrease the correlation score computed by the detector while maintaining the image quality. As a result, if a curve is below another one in a distortion vs. correlation plot, it means that the first attack has a stronger impact on the watermark than the second one. Looking at Figure 5, it is obvious that both proposed block replacement attacks outperform Gaussian blurring and JPEG compression. In particular, the correlation score drops below the detection threshold $\tau_{\text{detect}} = 1.5$ around 40 dB with block replacement attacks while it is necessary to introduce a distortion around 36 dB to obtain the same result with the reference attacks. Furthermore, assuming that the parameters of the attacks are set so that the introduced distortion is similar to the one due to the the embedding process (38 dB), block replacement attacks trap the detector while watermarks submitted to reference attacks can still be detected. In other terms, from an attacker perspective, the introduced block replacement strategy allows to improve the trade-off distortion vs. correlation in comparison with other standard reference attacks. Both block replacement attacks exhibit roughly the same performance. However, block projection on PCA-defined subspace requires fewer computations than adaptive number of blocks combination.

5. CONCLUSION

In some applications, digital watermarks are embedded to reduce the potential usages of protected data or to identify customers which have broken their license agreement. In such situations, users are likely to be willing to remove this hidden data which can be used against them. Thus, resistance to strong hostile attacks has to be considered and not only survival after *usual* signal processing operations since attackers are likely to introduce some intelligence in their attacks. In security fields, improvements usually come up from the competition between technology providers and attackers. In this paper, a novel attack inspired from fractal coding has consequently been proposed in an effort to anticipate the possible behavior of malicious users. The initial idea is to exploit the self-similarities of the signal to desynchronize the watermark while keeping the host data synchronized. More precisely, similar blocks carrying different watermark samples are exchanged to trap the detector. However, the successive improvements presented in the paper have turned the original attack which basically simulated fractal coding into an intra-signal collusion attack. Several similar blocks carrying alternative watermarks are isolated from the watermarked signal and combined to obtain a candidate block for replacement. Experiments have highlighted the efficiency of this attack with image documents in comparison with usual signal processing attacks such as Gaussian blurring and JPEG lossy compression. Complementary studies have also reported similar results when audio [20, 21, 30] or video [7, 8] signals are considered.

The attacker basically exploits the fact that the embedding algorithm does not consider the self-similarities of the signal. It is possible to build some sets of similar blocks which on the other hand are not assumed to carry similar watermark samples. This is a weak link of most watermarking schemes today and an informed attacker can exploit it to defeat the protection system. Now the question is: which countermeasures can be introduced by technology providers to disable, or at least decrease the impact, of such an attack? Intuitively, if *similar signal blocks carry similar watermarks*, the presented block replacement strategy is likely to be ineffective i.e. the introduced watermark has to be coherent with the self-similarities of the host signal. This can be seen as an intermediary specification between the security requirements for steganography -the embedded watermark should be statistically invisible [33] so that an attacker cannot even detect the *presence* of the hidden watermark- and the absence of any one for non-secure applications such as data hiding or broadcast monitoring. Unfortunately this intuitive statement does not give any clue on how to obtain such *coherent watermarks* in practice. An early study in video has demonstrated that security can be improved by making the watermark coherent with camera motion [8], so that temporal frame averaging after registration becomes useless. However, a generic approach has still to be found to solve the problem in the general case. In particular, this new specification raises many interesting questions. Is it possible to obtain such a coherent watermark for whatever host signal? If not, which strategy should be enforced to minimize the impact of block replacement based attacks? Then, how many bits can be reliably embedded? Does the achievable capacity depend on the host signal or not?

6. ACKNOWLEDGMENTS

This work has been supported in part by the European Commission through the IST Program under Contract IST-2002-507932 ECRYPT. Furthermore, the authors want to thank Darko Kirovski from Microsoft Research and Professor Ingemar Cox from University College London for fruitful discussions.

7. REFERENCES

[1] S. Byers, L. Cranor, D. Korman, P. McDaniel, and E. Cronin. Analysis of security vulnerabilities in the movie production and distribution process. In *Proceedings of the ACM Workshop on Digital Rights Management*, pages 1–12, October 2003.

[2] Certimark. http://www.certimark.org.

[3] Checkmark. http://watermarking.unige.ch/checkmark.

[4] H. Cohen. *A Course in Computational Algebraic Number Theory*. Springer-Verlag, 1993.

[5] I. Cox and M. Miller. Preprocessing media to facilitate later insertion of a watermark. In *Proceedings of the International Conference on Digital Signal Processing*, volume 1, pages 67–70, July 2002.

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

[7] G. Doërr and J.-L. Dugelay. New intra-video collusion attack using mosaicing. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, volume II, pages 505–508, July 2003.

[8] G. Doërr and J.-L. Dugelay. Secure background watermarking based on video mosaicing. In *Security, Steganography and Watermarking of Multimedia Contents VI*, volume 5306 of *Proceedings of SPIE*, January 2004.

[9] G. Doërr and J.-L. Dugelay. Security pitfalls of frame-by-frame approaches to video watermarking. *IEEE Transactions on Signal Processing, Supplement on Secure Media*, October 2004.

[10] DVD Copy Control Association. http://www.dvdcca.org.

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

[12] M. Gharavi-Alkhansari and T. Huang. A fractal-based image block-coding algorithm. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume V, pages 345–348, April 1993.

[13] F. Hartung, J. Su, and B. Girod. Spread spectrum watermarking: Malicious attacks and counterattacks. In *Security and Watermarking of Multimedia Contents*, volume 3657 of *Proceedings of SPIE*, pages 147–158, January 1999.

[14] A. Herrigel, S. Voloshynovskiy, and Y. Rytsar. The watermark template attack. In *Security and Watermarking of Multimedia Contents III*, volume 4314 of *Proceedings of SPIE*, pages 394–405, January 2001.

[15] M. Holliman, W. Macy, and M. Yeung. Robust frame-dependent video watermarking. In *Security and Watermarking of Multimedia Contents II*, volume 3971 of *Proceedings of SPIE*, pages 186–197, January 2000.

[16] A. Jacquin. A novel fractal block-coding technique for digital images. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume IV, pages 2225–2228, April 1990.

[17] I. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 1986.

[18] JPEG Standard. Digital compression and coding of continuous-tone still images. In *JTC1/SC29/WG1 10918-1*. ISO/IEC, February 1994.

[19] D. Kirovski and H. Malvar. Robust covert communication over a public audio channel using spread spectrum. In *Proceedings of the Fourth International Workshop on Information Hiding*, volume 2137 of *Lecture Notes on Computer Science*, pages 354–368, April 2001.

[20] D. Kirovski and F. Petitcolas. Blind pattern matching attack on watermarking systems. *IEEE Transactions on Signal Processing*, 51(4):1045–1053, April 2003.

[21] D. Kirovski and F. Petitcolas. Replacement attack on arbitrary watermarking systems. In *Proceedings of the ACM Digital Rights Management Workshop*, volume 2696 of *Lecture Notes on Computer Science*, pages 177–189, July 2003.

[22] M. Kutter. Watermarking resisting to translation, rotation and scaling. In *Multimedia Systems and Applications*, volume 3528 of *Proceedings of SPIE*, pages 423–431, November 1998.

[23] M. Kutter and F. Petitcolas. A fair benchmark for image watermarking systems. In *Security and Watermarking of Multimedia Contents*, volume 3657 of *Proceedings of SPIE*, pages 226–239, January 1999.

[24] G. Langelaar, R. Lagendijk, and J. Biemond. Removing spatial spread spectrum watermarks by nonlinear filtering. In *Proceedings of the European Signal Processing Conference*, volume IV, pages 2281–2284, September 1998.

[25] G. Øien, S. Lepsøy, and T. Ramstad. An inner product space approach to image coding by contractive transformations. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume III, pages 2773–2776, May 1991.

[26] OpenWatermark. http://www.openwatermark.org.

[27] Optimark. http://poseidon.csd.auth.gr/optimark.

[28] J. Ó Ruanaidh and T. Pun. Rotation, scale and translation invariant digital image watermarking. *Signal Processing*, 68(3):303–317, May 1998.

[29] F. Petitcolas, R. Anderson, and M. Kuhn. Attacks on copyright marking systems. In *Proceedings of the Second International Workshop on Information Hiding*, volume 1525 of *Lecture Notes on Computer Science*, pages 219–239, April 1998.

[30] F. Petitcolas and D. Kirovski. The blind pattern matching attack on watermarking systems. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume IV, pages 3740–3743, May 2002.

[31] C. Rey. *Tatouage d'Images: Gain en Robustesse et Intégrité des Images*. PhD thesis, Université d'Avignon, France, February 2003.

[32] C. Rey, G. Doërr, J.-L. Dugelay, and G. Csurka. Toward generic image dewatermarking? In *Proceedings of the IEEE International Conference on Image Processing*, volume III, pages 633–636, September 2002.

[33] P. Sallee. Model-based steganography. In *Proceedings of the Second International Workshop on Digital Watermarking*, volume 2939 of *Lecture Notes in Computer Science*, pages 154–167, October 2003.

[34] Secure Digital Music Initiative. http://www.sdmi.org.

[35] Stirmark. http://www.petitcolas.net/fabien/watermarking/stirmark.

[36] J. Su and B. Girod. Power-spectrum condition for energy-efficient watermarking. *IEEE Transactions on Multimedia*, 4(4):551–560, December 2002.

[37] S. Voloshynovskiy, S. Pereira, A. Herrigel, N. Baumgärtner, and T. Pun. Generalized watermarking attack based on watermark estimation and perceptual remodulation. In *Security and Watermarking of Multimedia Contents II*, volume 3971 of *Proceedings of SPIE*, pages 358–370, January 2000.

[38] S. Voloshynovskiy, S. Pereira, V. Iquise, and T. Pun. Attack modeling: Towards a second generation watermarking benchmark. *Signal Processing*, 81(6):1177–1214, June 2001.

[39] Watermark Evaluation Testbed (WET). Contact Professor E. Delp, Purdue University, USA.

[40] M. Wu and B. Liu. Attacks on digital watermarks. In *Proceedings of 33th Asilomar Conference on Signals, Systems, and Computers*, volume II, pages 1508–1512, October 1999.