EURASIP Journal on Applied Signal Processing 2004:14, 1–14
© 2004 Hindawi Publishing Corporation

Video Waterscrambling: Towards a Video Protection Scheme Based on the Disturbance of Motion Vectors

Yann Bodo
TECH/IRIS/CIM, France Telecom R&D, 4 rue du Clos Courtel, 35512 Cesson Sévigné Cedex, France
Email: yann.bodo@wanadoo.fr

Nathalie Laurent
TECH/IRIS/CIM, France Telecom R&D, 4 rue du Clos Courtel, 35512 Cesson Sévigné Cedex, France
Email: nathalie.laurent@francetelecom.com

Christophe Laurent
TECH/IRIS/CIM, France Telecom R&D, 4 rue du Clos Courtel, 35512 Cesson Sévigné Cedex, France
Email: christophe2.laurent@francetelecom.com

Jean-Luc Dugelay
Multimedia Communication Department, Institut EURECOM, 2229 Route des Cretes, BP 193,
06904 Sophia-Antipolis Cedex, France
Email: jean-luc.dugelay@eurecom.fr

Received 31 March 2003; Revised 19 December 2003

With the popularity of high-bandwidth modems and peer-to-peer networks, the contents of videos must be highly protected from piracy. Traditionally, the models utilized to protect this kind of content are scrambling and watermarking. While the former protects the content against eavesdropping (a priori protection), the latter aims at providing a protection against illegal mass distribution (a posteriori protection). Today, researchers agree that both models must be used conjointly to reach a sufficient level of security. However, scrambling works generally by encryption resulting in an unintelligible content for the end-user. At the moment, some applications (such as e-commerce) may require a slight degradation of content so that the user has an idea of the content before buying it. In this paper, we propose a new video protection model, called *waterscrambling*, whose aim is to give such a quality degradation-based security model. This model works in the compressed domain and disturbs the motion vectors, degrading the video quality. It also allows embedding of a classical invisible watermark enabling protection against mass distribution. In fact, our model can be seen as an intermediary solution to scrambling and watermarking.

Keywords and phrases: content protection, video scrambling, watermarking, motion estimation.

1. INTRODUCTION

With the fast proliferation of high-bandwidth personal modems (especially ADSL and cable modems), the exchange of digital multimedia contents has drastically increased. This exchange is also greatly facilitated by the emergence of digital communities that share many files across peer-to-peer networks. Among these shared files, many are copyrighted, and in this context, it is necessary to control their distribution in an open network such as the Internet.

This occurred recently with the MP3 revolution in digital audio contents. From this time, many MP3 processing softwares including CD rippers, MP3 encoders, and MP3 players have been posted for free on the Web allowing end-users to build their own MP3 record collections from their own CDs. Inevitably, this situation has caused an incredible piracy activity and Web sites have begun to stream and provide copyrighted MP3 music for free. In response to this piracy situation, the Recording Industry Association of America (RIAA) created the Secure Digital Music Initiative (SDMI, http://www.sdmi.org) working group to explore technologically secure alternatives to the MP3 format. This group aimed at protecting online music from illegal duplication and mass distribution. To test the proposed solutions, on September 6th 2000, it issued the SDMI challenge to the digital community inviting people to crack their system. Unfortunately, students from the Princeton University successfully hacked the SDMI technology.

This digital audio situation shows that with the digital era comes the need for the adaption of business practices. Traditional methods are often not successful when implemented online. However, technology can evolve drastically faster than the business world and it has become increasingly difficult for the entertainment industry to adapt at the same rate as the fast changing world of digital innovations. The proposal of digital rights management (DRM) technology has been initiated in an attempt to overcome these problems and to initiate new working practices. The DRM systems generally provide two essential functions: management of digital rights by identifying, describing, and setting the rules of content usage, and digital management of rights by securing the content and enforcing usage rules. However, a recent report from the Commission of European Communities [1] shows that today, DRM systems are neither widely deployed nor widely accepted, mainly due to the reduction of ease of use, the prevention of generally accepted uses (e.g., private copy of content), and a lack of flexibility and interoperability between existing systems. Therefore, while piracy practices are very active, new digital businesses are not able to take place and peer-to-peer communities can still occur.

In fact, two different problems arise: the content protection and copy protection. While content protection aims at protecting the content itself against eavesdropping, the objective of copy protection is to avoid the illegal mass distribution of copyrighted contents.

Content protection is an old issue in the digital TV environment and new cryptographic tools have been proposed [2], namely *conditional access*, in this context. Conditional access systems work by scrambling the content, that is, encrypting the content with keys that change frequently. This kind of protection has been adopted by all digital TV broadcasting standards, such as digital video broadcasting (DVB) in European countries.

The problem of copy protection has been tackled in the analogue TV world by Macrovision with the proposal of the APS copy protection scheme based on the differences in the way VCRs and TVs operate. However, copy protection in the analogue world is of limited importance due to the degradation of video quality along with the copy generations. Conversely, this issue is crucial in the digital environment in which digital content can be cloned without loss of quality. In this way, CD technology has been the first victim with the advent of CD writers. Consequently, in 1996, the Motion Picture Association of America (MPAA), the Consumers Electronics Manufacturers Association (CEMA), and members of the computer industry put together an *ad hoc* group called Copy Protection Technical Working Group (CPTWG) to discuss the technical problems of protecting digital video from piracy, particularly in the domain of digital versatile disk (DVD) [3]. This working group has addressed four key problems: content protection, analogue copy protection, digital copy generation management, and exchange of contents across digital networks. Unfortunately, the content-scrambling-system (CSS), chosen to encrypt the DVD content, used a weak 40-bit key and the algorithm was quickly hacked into by Stevenson [4] in 1999, making it possible to extract the contents of DVDs in unscrambled form.

All these aforementioned facts show that the content protection against eavesdropping and illegal copy is a challenging task and we cannot always be sure that a proposed method will be totally secure. On the other hand, there is a difficult tradeoff between system complexity and cost. In fact, manufacturers often accept a limited amount of piracy by adopting the well-known mantra “keeping honest people honest.”

Among methods proposed in literature to protect video contents, two approaches are classically utilized: scrambling and watermarking. As scrambling is generally based on old and proven cryptographic tools [5], it efficiently ensures confidentiality, authenticity, and integrity of messages when they are transmitted over an open network. However, it does not protect against unauthorized copying after the message has been successfully transmitted and decrypted [6]. This kind of protection can be handled by watermarking [7], which is a more recent topic that has attracted a large amount of research and is perceived as a complementary aid in encryption. A digital watermark is a piece of information inserted and hidden in the media content. This information is imperceptible to a human observer but can be easily detected by a computer. Moreover, the main advantage of this technique concerns the nonseparability of the information to be hidden and the content. A watermark system consists of an embedding algorithm and a detector function. The embedding algorithm inserts a message inside media, the detector function is then used to verify the authenticity of the media by detecting the mark. The most important properties of a watermarking scheme include [8] robustness, fidelity, tamper resistance, and payload. More details regarding the common watermarking properties can be found in various papers, such as [8, 9]. Finally, a wide number of watermarking technologies have been developed and deployed today for a wide variety of applications as discussed in [8].

In this paper, we present an alternative video protection model that we call *waterscrambling*. This new model is motivated by the following observations:

(i) a scrambling-based protection scheme totally prevents the end-user from seeing the content. However, it can be useful, for some applications such as e-commerce, to show the content under a degraded form in order to provoke an impulsive buying action;

(ii) a video protection solution based solely on a watermarking approach does not prevent the propagation of the content. The watermark must be coupled with another secure scheme to prevent illegal copy. In fact, a watermark-based video protection scheme needs a watermark compliant video player in order to be effective.

Our waterscrambling solution can be seen as an intermediary solution between scrambling and watermarking. By disturbing the video sequence motion vectors in compressed form, our approach degrades the video quality, but still enabling video content to be perceived by an end-user, giving him an idea of the original content. In this sense, our approach is a scrambling variant. By also being able to embed invisible information in the motion vectors, our approach satisfies the previously recalled watermarking requirements.

After presenting an overview of the classical video protection schemes in Section 2, our waterscrambling process will be detailed in Section 3. Finally, our conclusions and perspectives will be discussed in Section 4.

2. AN OVERVIEW OF VIDEO CONTENT PROTECTION SCHEMES BASED ON WATERMARKING TECHNIQUES

2.1. Video content protection problem statement

As underlined in the previous section, two different problems have to be considered when protecting video content: the protection of the content itself and the prevention of illegal copy.

Content protection is an old issue in the digital TV area and works by scrambling (i.e., encrypting) video content [2]. To achieve a sufficient security level, due to the huge amount of data giving rise to specific attacks, the heart of the scrambling security is a combination of a proven encryption algorithm with a frequent change of keys.

Obviously, the topic of content protection has also been discussed in the CPTWG with the aim of protecting the content of DVDs [3]. For this purpose, the CSS algorithm developed by Matsushita has been adopted.

The CPTWG has also considered the copy protection problem by embedding a pair of bits in the header of the MPEG stream. This protection scheme, called copy generation management system (CGMS), encodes one of the three possible rules for copying: “copy freely” (i.e., the video may be freely copied), “copy never” (i.e., the video may never be copied), and “copy once” (i.e., only a first generation copy is authorized).

The arrival of new digital networks, thanks to powerful high-bandwidth digital buses such as IEEE1394, also needs new security specifications. In these networks, all devices are connected through digital links and it must be ensured that video content is not transported in clear text, nor can be illegally copied during its transfer between devices. This problem is generally resolved by providing a mechanism that strongly authenticates all network devices and that allows the content encryption key exchange between authenticated devices. Today, two competing solutions tackle this kind of problem: the oldest one is digital transmission content protection (DTCP) that was developed by 5C (a consortium of five companies, including Hitachi, Intel, Matsushita, Sony, and Toshiba). The second solution, SmartRight, was designed by Thomson Multimedia and is today supported by eight other companies (Canal+ Technologies, Nagravision, Gemplus, SchlumbergerSema, ST, Pioneer, Micronas, and SCM Microsystems). The protection of video content over digital networks is of prime importance and for this reason, the CPTWG has created the Digital Transmission Discussion Group (DTDG) to explore the issue.

2.2. Watermarking protection schemes

A video watermark technique consists of the hiding of information into a video sequence to protect the video content as a whole. One way of embedding a watermark into a video is to independently mark all the video frames by using techniques from the still image watermarking area. Another way is to use the temporal information of the video. Consequently, we can classify video watermarking schemes into two main categories: still image-based techniques and video-adapted techniques. Today, most of the video watermarking approaches rely on the extension of still image algorithms. However, these algorithms generally lack robustness since they do not fully consider the video temporal axis. Literature has provided few watermarking algorithms that consider temporal information as a key advantage to propose a more robust solution. Effectively, it seems natural to consider that the robustness of a watermark can be greatly improved by considering the following two video properties.

(i) *Information amount*. A video sequence represents a larger amount of information than a still image. Therefore, the insertion space of the watermark is increased and can be exploited to insert a more robust mark.

(ii) *Motion information*. The object motion increases the visibility of the mark.

However, the insertion of the watermark is also constrained by the following.

(i) *Runtime complexity*. The complexity of the mark insertion scheme should be small, and ideally, the algorithm should run in real time.

(ii) *Compression constraint*. The mark embedding process should not produce a compressed marked bitstream larger than the unmarked one.

(iii) *New class of attacks*. Video watermarking leads to somewhat different attacks than those used in image watermarking. Moreover, the mark should be detectable even after a loss of synchronization due to temporal subsampling or to the selection of a subsequence.

2.2.1. Still image-based techniques

Primarily, watermarking algorithms for video were simply an adaptation of still image techniques. Langelaar et al. [10], Nikolaidis and Pitas [11], and O’Ruanaidh et al. [12] have each proposed good overviews of still image watermarking techniques that can be used to mark video content if we consider the video sequence as a succession of independent still images. To embed a watermark, we can work in the spatial domain or in a transform domain. In the same way, we can work with compressed or original uncompressed data. Finally, to increase the invisibility of the inserted mark, researchers often use a psychovisual mask. Effectively, regarding the properties of the human visual system (HVS) increase the energy of the watermark without generating additional visual artifacts. Naturally, these possibilities are also valid when marking video, explaining why many still image watermarking concepts are directly used in video.

One of the main techniques used in watermarking is the spread spectrum approach firstly introduced by Cox et al. [13]. In this approach, the use of a pseudorandom bit generator (PRBG) modulated with an oversampling version of the mark allows to generate redundancy and randomness in the embedding process, resulting in a largely increased robustness. Based on this technique, an approach working in the spatial domain has been proposed by Hartung and Girod [14]. In this method, the video is considered as a 1D signal. However, the authors do not really consider the intrinsic properties of the video because they store the video signal into a 1D vector, loosing thus the spatial information of the still frame as well as the temporal information that characterizes a video. It has to be noted that this scheme is among the first to deal with video watermarking.

Among watermarking approaches working in the compressed domain, 8 × 8 blocks are generally employed when embedding the watermark due to their use in compression standards such as MPEG or JPEG. Koch and Zhao [15] have developed a still image watermarking method using the JPEG compression scheme and working in the frequency domain. They first apply a discrete cosine transform (DCT) on luminance blocks before quantizing them. Then, they pseudorandomly select three of these quantized coefficients in the medium frequencies over which they apply an insertion rule. This consists of imposing a pattern rule onto the three coefficients depending on the bit to be embedded. Dittman et al. [16] have proposed two watermarking algorithms: one adapted from this block-based technique and the other from the algorithm developed by Fridrich in [17]. In the first approach, the embedding is performed by marking the 8 × 8 blocks in the DCT frequency domain. In the second one, the authors embed the mark in the spatial domain. The main advantage of the second approach is that it is able to embed more than 250 bits and to withstand stirmark attack. The first algorithm is more suitable for video and improves the video quality. However, its complexity does not allow for the real-time constraint. It has to be noted that both approaches use the HVS properties to increase the robustness and the invisibility of the mark.

In [18], Wolfgang et al. proposed a still image watermarking scheme that they have adapted to video content by embedding the mark in the intra frames. In this work, the authors work in the DCT domain and use a spatial masking approach.

One of the well-known techniques proposed in the video watermarking topic is the just another watermarking system (JAWS) algorithm developed by Kalker et al. that has been firstly designed for broadcast monitoring [19]. JAWS is based on simple operations allowing for the real-time requirement in which the video is considered as a succession of still images. The watermarking payload was initially one bit, but in [20], the authors achieved an embedding of 36 bits/s thanks to the symmetrical phase only matched filtering (SPOMF) algorithm. This improvement was presented in [19] where the authors generate a pseudorandom pattern according to the message to be embedded. The watermark is then perceptually shaped and scaled before being inserted. Although this algorithm is based on still image techniques, it shows a good robustness and is today one of the major algorithms proposed in video watermarking. It is useful to note that the JAWS-based watermarking solution is proposed by Philips under the commercial name of *Watercast*.[1]

Another powerful commercial solution is the one provided by Nextamp.[2] Their algorithm is mainly based on the still image watermarking scheme developed by Koch and Zhao [21] and the approach proposed by Baudry et al. [22]. This algorithm meets the real-time constraint.

2.2.2. *Video adapted technique*

Langelaar et al. [10] and Doërr and Dugelay [23] have proposed comprehensive overviews of video watermarking techniques. While the former deals with basic approaches, the latter proposes a good view of the actual watermarking problem. If we consider the problem of digital broadcasting, for which the runtime complexity must be drastically reduced, there is a need of getting an algorithm that meets the real-time constraint and that embeds the mark in the compressed domain. Hartung and Girod [14] proposed a method that works in the compressed domain and that involves the embedding of the mark in the video intra frames. Then, they made a drift compensation for visibility purposes. The mark is transformed into a 2D signal before being embedded in the image. The authors consider the compression problem but they do not use the motion at all. Now, it seems natural to employ the motion data since it embeds a high-value added information into the intrinsic video content. For this purpose, some techniques are based on temporal 3D transforms [24, 25], while others use motion vectors obtained by a motion estimator [26, 27]. However, it must be emphasized that 3D approaches generally consider temporal dimension in the same way as spatial ones, although they do not hold the same kind of information. The same drawback has been noted in the source coding field where this kind of approach did not reach good results. In [25], Tewfik et al. use a temporal wavelet transform in order to identify the low and fast motion areas in the video. They first extract the different scenes of the video by applying a temporal segmentation, and then apply their watermarking algorithm. By doing so, they can embed two different watermarks depending on the motion activity, and then adapt the watermark to the content. The temporal axis is performed here for discriminating the content and not to embed the mark. Moreover, the temporal wavelet transform greatly increases the complexity of the algorithm. In [24], a 3D discrete Fourier transform (DFT) is used. Due to the separability property of this transform, it can be considered as the composition of a 2D spatial DFT and a 1D temporal DFT. The mark is embedded in the magnitude component of the DFT coefficients.

[1] http://www.watercast.com.
[2] http://www.nextamp.com.

Although the temporal aspect is used, the complexity of this design is a drawback. Most of the temporal transforms are processed in order to discriminate between the different characteristics of the video content. Most of the time, the discrimination is performed on a motion basis (static/dynamic area) and/or feature basis (edge/texture area) resulting in a high computational cost. Some research works deal with watermarking schemes using motion vectors to embed the mark. This approach seems to be more appropriate to this media, but the preferred method is the inclusion of a psychovisual mask in order to separate the dynamic zones from the static ones. Marking motion vectors was first introduced by Jordan et al. in [26], in which the authors select a set of motion vectors over which they apply a parity rule to embed the mark. Later, Zhang et al. [27] used this principle and adapted the insertion rule by selecting the vector components that have the greatest magnitude. In [28], Lancini et al. embed the mark in the spatial domain. They first design a mask composed of three different components, one for luminance masking, one for texture masking, and the third for temporal masking, then they apply a classical spread-spectrum technique to embed the mark as in [13]. As mentioned in [29], JAWS is one of the main algorithms available to protect and control the illegal copy of DVDs. The main constraint discussed in this DVD protection topic is the real-time constraint, needed for the detection algorithm in order to be incorporated in the DVD decoding process. To reach this goal, the detection is performed directly on the MPEG stream resulting in a drastic reduction of the complexity at the cost of a slight reduction in performance. Finally, an adaptation of the techniques present in JAWS was designed in [30], resulting in a new algorithm that can be used to protect the digital cinema area. In this last design, the temporal axis is the only one used due to different constraints. Indeed, the handled camera used to make a screener of the projected movie introduces filtering and serious geometrical distortions. Thus, in order to be resistant to geometric attacks, they adapt their technique to mark only the temporal axis. More recently, with the growth of the MPEG4 standard, some watermarking algorithms have been designed to protect the MPEG4 objects. In [31, 32], the procedure consists first of extracting the objects from the stream and then embedding a watermark to protect each of these objects.

In conclusion, transform domain algorithms make the watermarking algorithms complex and thus costly. For broadcasting applications, real time is necessary. The best method in this context is to embed and detect in the compressed domain, or in the spatial (or temporal) domain.

Finally, it can be noted that some authors have proposed hybrid methods to protect digital contents by combining cryptography and watermarking. In this way, Macy et al. use in [33] a multilevel scrambling approach together with watermarking: the video content scrambling is based on the disturbance of DCT coefficients and the watermarking performs a classical spread spectrum in the spatial domain. In the same way, Bao [34] proposes to mix public key cryptography and watermarking. In [35], Cheng and Li perform a partial encryption of the content by using a wavelet transform and a quadtree data structure. More recently, Zeng and Lei [36] have proposed a video protection technique combining a selective bit scrambling scheme in the frequency domain, block shuffling, and block rotation of the transform coefficients and motion vectors.

3. THE VIDEO WATERSCRAMBLING APPROACH

3.1. Introduction

Until now, most watermarking systems were designed to protect a media content by inserting a robust and invisible copyright mark. Our approach is slightly different, since we use watermarking techniques to insert a visible mark thus “scrambling” the video content. As underlined in the previous section, scrambling is commonly employed to prevent unauthorized access to video data and works by distorting the data such that the video appears unintelligible to a viewer.

In our mind, this kind of approach is the most effective one to protect the video against eavesdropping. However, in some cases, it can be useful to show the video content under a degraded form until the end-user subscribes to the corresponding service. In fact, a video protection scheme that gives the user an idea of the content can lead to impulsive subscription action, more than a pure scrambling approach.

In this section, we propose such a scheme that we call *video waterscrambling*. Contrary to classical scrambling systems, our process distorts the video quality and is able to regulate the video visibility from the original to unintelligible quality. Moreover, it does not disturb the video statistics as much as other schemes and it is not difficult to keep a good compression ratio by tuning the waterscrambling level. Finally, contrary to most existent watermarking and scrambling techniques, our waterscrambling system can run in real time during an MPEG compression phase (because it uses motion vectors computed during the compression process) or after the compression by extracting motion vectors from the MPEG bitstream.

Few research works have proposed adjustable video quality schemes for security purposes. An access control system based on fractal coding theory was proposed in [37]. The authors use a fractal coding scheme to adaptively and partially encrypt an image. In fact, they present an approach based on iterated function system coding (IFSC) providing both compression and hierarchical access control for images at various resolution levels. This hierarchical access control scheme allows the terminals to display an image at a low resolution level. The higher resolution levels (which correspond to a better image quality) are displayed according to the receiver access rights that are usually determined by the subscription agreement.

In our approach, the distortion level is more flexible. Effectively, contrary to [37] that proposes a coarse granularity by using only eight encryption levels, we propose a scheme with fine and continuous granularity. Moreover, our process is easy to implement and runs in real time. In order to reach this fine granularity, we build a visible marking system based on the use of the video motion vectors. As mentioned in

Section 2.2.2, only two important watermarking techniques based on motion vectors have been proposed in literature [26, 27]. However, both methods suffer from serious drawbacks. The approach presented in [26] is based on the parity of the motion vector components which is not robust. Effectively, filtering can destroy their watermarks by changing the parity of some motion vectors. Moreover, both methods are not reversible, which becomes a problem when the reconstruction of the original video is needed as for our concept. Thus, the goal of the waterscrambling approach proposed in this paper consists in finding a reversible "pseudoscrambling" solution which uses and modifies the MPEG motion vectors. However, if our major idea consists of designing a new kind of a pseudoscrambler, another interest of this approach concerns the possibility of inserting a watermark during the scrambling process in real time (allowing us to build a complete protection system). To anticipate this, our waterscrambling solution must be compatible with a watermarking solution. In fact, the insertion rule of our system must resist manipulations usually performed on video data (e.g., compression, filtering, etc.). The marked motion vectors must be maintained in a local space determined by the insertion rule to resist attacks aiming at displacing them around their initial position.

3.2. *Embedding scheme*

Our waterscrambling procedure is included in an MPEG compression scheme. The first step consists of extracting the motion vectors to be marked and two different approaches can be envisaged for this purpose. The first method uses a syntactic analyzer to extract motion vectors from the MPEG compressed bitstream, and in this case, the waterscrambling system is an independent module. The second one consists of directly modifying motion vectors to be waterscrambled during the MPEG compression scheme. In this case, we must use a module compliant with the standard compression one.

To waterscramble the video, a visible mark, defined by a binary vector $W \in \{-1, 1\}^N$ (where N denotes the size of the mark) is added to a set of chosen motion vectors. In order to increase the robustness of the mark, we apply a permutation $\sigma_f(W)$ on W at each frame f. First of all, as proposed in [13] and by analogy to spread spectrum communications, the mark is spread over many frequency bins so that the energy in each one is very small. Thus, we extract from each frame f corresponding to an MPEG P or B frame, the set of the m_f motion vectors denoted by $V_f = \{\vec{d}_f^{\,i}, 1 \leq i \leq m_f\}$. Then, a set $\widetilde{V}_f$ ($\widetilde{V}_f \subseteq V_f$) of $k_f \leq m_f$ selected motion vectors is used to superpose the digital mark signal $\sigma_f(W)$ onto the original signal of the selected motion vectors:

$$\forall \vec{d}_f = (d_f^x, d_f^y)^T \in \widetilde{V}_f, \quad \vec{d}_f^{\,W} = \vec{d}_f + \Phi(\alpha, \sigma_f(W), K_{\sigma_f(W)}), \tag{1}$$

where $\vec{d}_f$ is a motion vector belonging to $\widetilde{V}_f$, $\vec{d}_f^{\,W}$ is the resulting marked motion vector, α denotes the mark strength (which could be different in various data samples), and Φ is a reversible function depending on W and K, K being a waterscrambling secret key that may be used to enforce security.

To determine the set $\widetilde{V}_f$ of chosen motion vectors, we use the waterscrambling key $K_{\sigma_f(W)}$ to initialize a pseudorandom number generator (PRNG) which outputs k_f indexes k_f^i ($i \in [1, m_f]$) denoting the indexes of the motion vectors of interest in V_f: $\widetilde{V}_f = \{\vec{d}_f^{\,j},\ j \in \{k_f^i\}_{i\in[1,m_f]}\}$.

We point out that a PRNG is a cryptographic algorithm used to generate numbers that must appear random [5]. It has a secret state and it must generate outputs that are indistinguishable from random numbers to an attacker who does not know and cannot guess the secret state. In this sense, it is very similar to a stream cipher. Additionally, a PRNG must be able to alter its secret state by processing input values that may be unpredictable. A PRNG often starts in a guessable state and must process many inputs to reach a secure state. Yarrow [38] is an example of a secure PRNG that can be used in our waterscrambling scheme because it has been proven to be more robust than other PRNGs. The major design principle of the Yarrow system is that its components are more or less independent, so that systems with various design constraints can still use the general Yarrow design. The use of algorithm-independent components in the top level design is a key concept in Yarrow. The goal is not to increase the number of security primitives that a cryptography system is based on, but to leverage existing primitives as much as possible. Hence, Yarrow relies on one-way hash functions and block ciphers cryptographic primitives.

To enforce the security level, the waterscrambling key $K_{\sigma_f(W)}$ is changed for each new video to be waterscrambled. This key represents the initial state of the PRNG. Our waterscrambling system uses this secret waterscrambling key in a similar way to a symmetric cipher, that is, the key must be shared between the content provider and the end-user to enable the "dewaterscrambling" process. Consequently, a secure channel must be set up between both parties to securely transfer this key.

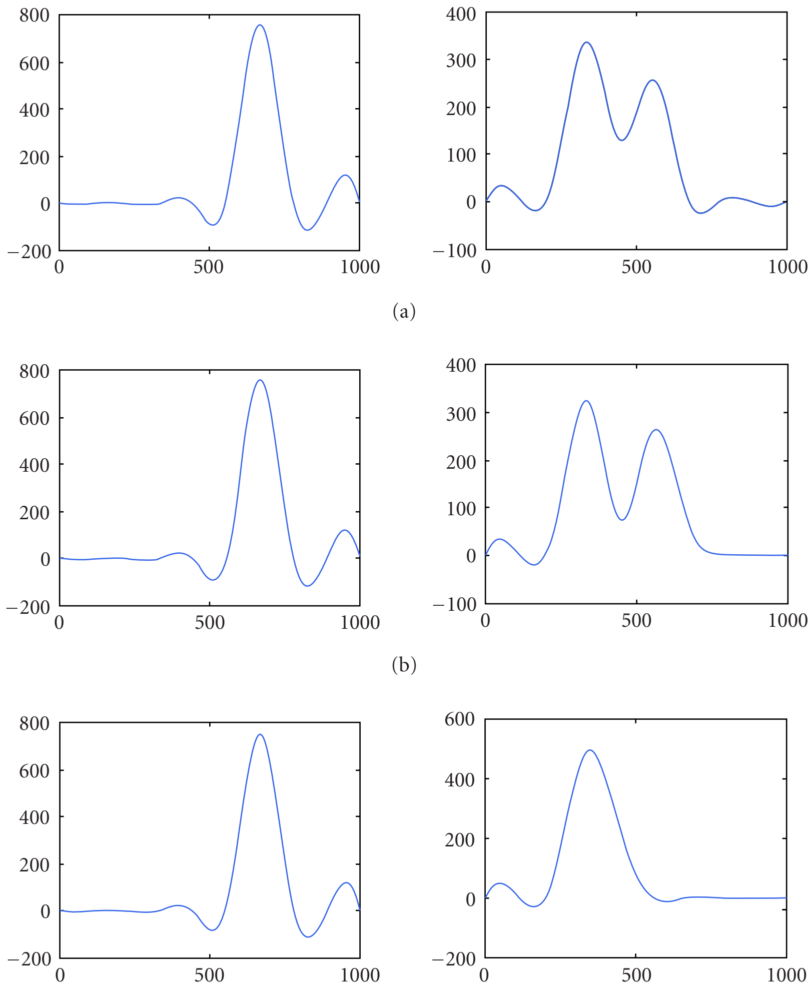
Moreover, as the "dewaterscrambling" process needs to know the strength α that the provider used to waterscramble the video, the key $K_{\sigma_f(W)}$ must be decomposed into two parts: the seven first bits of the key contain the strength α and the other bits denote the key itself used to initialize the PRNG.

The following waterscrambling embedding rule is then used:

$$\forall \vec{d}_f = (d_f^x, d_f^y)^T \in \widetilde{V}_f,$$

$$\vec{d}_f^{\,W} = \begin{cases} d_f^{W,x} = \begin{cases} d_f^x + \alpha \times \Upsilon(\sigma_f(W), K_{\sigma_f(W)}), & \text{if } \sigma_f^i(W) = +1, \\ d_f^x, & \text{otherwise}, \end{cases} \\ d_f^{W,y} = \begin{cases} d_f^y + \alpha \times \Upsilon(\sigma_f(W), K_{\sigma_f(W)}), & \text{if } \sigma_f^i(W) = -1, \\ d_f^y, & \text{otherwise}, \end{cases} \end{cases} \tag{2}$$

where $\sigma_f^i(W)$ denotes the ith component of the vector $\sigma_f(W)$ and W the visible mark. We can thus see that W is scattered in the image by embedding only one bit in each chosen motion vector of $\widetilde{V}_f$.



(c)

Figure 1: Modification of motion vectors distribution after the application of the waterscrambling. (a) Distribution of the x component (right) and y component (left) of the original motion vectors. (b) Modification of the distribution with a waterscrambling strength $\alpha = 20$. (c) Modification of the distribution with a waterscrambling strength $\alpha = 100$.

Υ is in this case a reversible function allowing for the tuning of the degree of quality degradation.

Finally, to spread the waterscrambling effect, we can insert the visible mark in the transform domain instead of the spatial one. For this purpose, we perform two 1D DCTs, the first one on the x components and the second one on the y components of a global vector $V = (V^x, V^y)^T \in \mathbb{R}^{2k_f}$ with $V^x = (d^x_{f_1}, d^x_{f_2}, \dots, d^x_{f_{k_f}})$ and $V_y = (d^y_{f_1}, d^y_{f_2}, \dots, d^y_{f_{k_f}})$.

By working in the transform domain, we are able to control the global energy added to the motion vectors by, for example, only disturbing high or middle frequencies. Moreover, we are able to keep the statistics of the motion vectors distribution, thus avoiding to increase the compression ratio. To reach this goal, we can define the function Υ such that it corresponds to a pseudohomothetic deformation of the coded motion vectors distribution (see Figure 1). Figure 1 shows an example of such motion vectors distribution. The distribution of the original motion vectors is illustrated on Figure 1a in which the left graph (resp., the right graph) shows the distribution of the x (resp., y) components. The modifications of these distributions with a waterscrambling strength $\alpha = 20$ and $\alpha = 100$ are, respectively, shown on Figures 1b and 1c. As it can be noted, protecting a video with a strength $\alpha = 20$ does not significantly change the distributions, thus allowing it to maintain approximatively the same compression ratio while degrading sufficiently the video quality. Conversely, for a strength $\alpha = 100$, the distribution of vector amplitudes in the y direction is greatly affected and the compression ratio is consequently degraded. Although most video codecs code motion vectors using a differential approach, these curves show nevertheless that the coding cost does not change significantly. Indeed, the motion vectors are slightly modified by the waterscrambling scheme and thus remain in a restricted

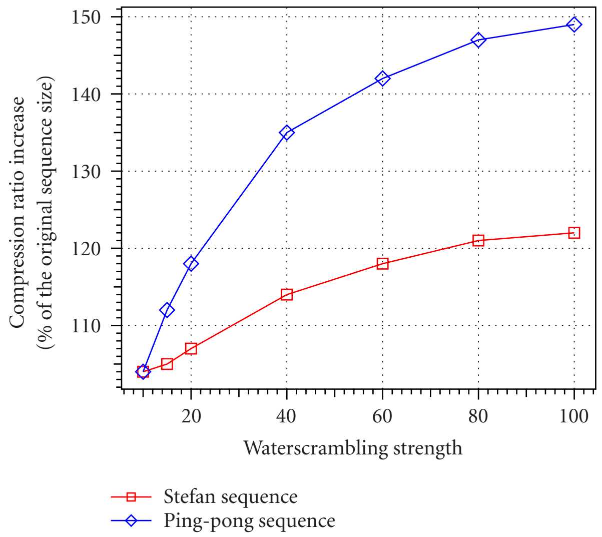


Figure 2: Variation of compression ratios according to the waterscrambling strength on Stefan and Ping-pong sequences.

spatial area. Consequently, by keeping approximately the same distribution of motion vectors, we ensure that the coding cost is close to be the same as the original video content. In the case where the compression ratio must remain the same as the original compressed video, we have to choose the function Υ adequately. In addition, Figure 2 shows the compression ratio variation according to the waterscrambling strength applied to the video. As mentioned before, we can note that a strength $\alpha = 100$ increases the compression ratio of about 35% when averaged over two video sequences. However, a strength $\alpha = 20$ generally suffices to degrade the video quality while keeping a sufficient level of visibility (see the second row of Figure 6). In this case, the increase of compression ratio is only of 10%, which is largely acceptable.

Once the visible mark is inserted, a classical watermarking approach can follow. A mix of scrambling and watermarking was first proposed in [39] in which two alternatives are presented. The first one embeds the watermark before scrambling the content. In this way, the content receiver descrambles only the content and the mark remains. The second alternative proposes to send a scrambled video without embedding a watermark. At the receiver side, the content is descrambled and conjointly watermarked.

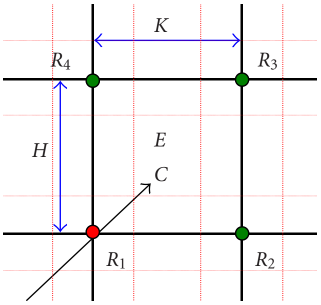


Figure 3: Construction of a reference grid to embed a watermark on motion vectors.

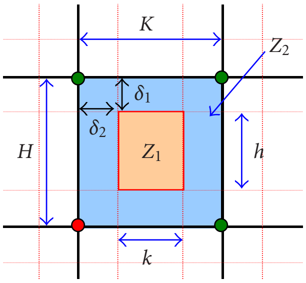


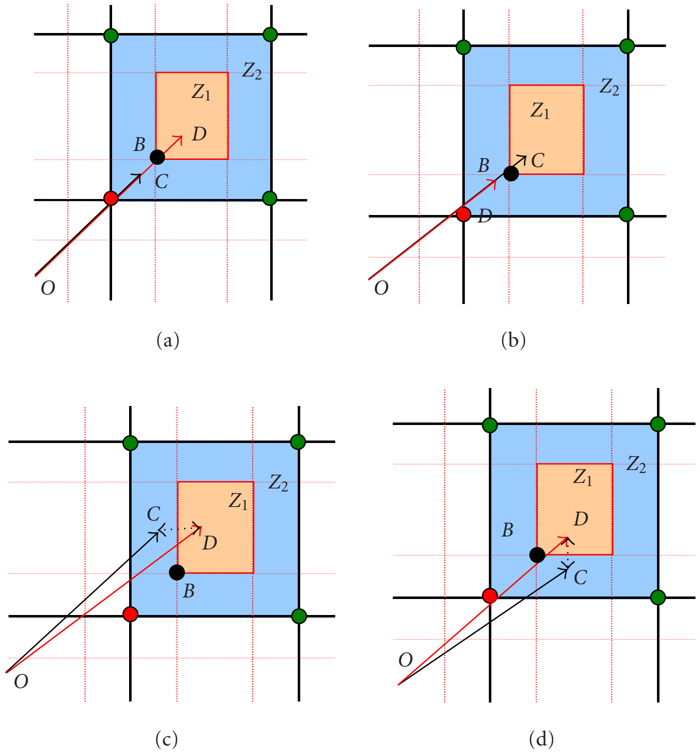
Figure 4: Block element partitioning to embed the mark.

In our process, both approaches can be envisaged. Effectively, we can add an invisible watermark W' on the same chosen motion vectors. Converse to the waterscrambling approach, the strength α may be chosen in order to maintain the invisibility of the mark W' onto the original signal. This watermarking process can be performed in the compressed or in the uncompressed domain. In our case, we have chosen to work in the uncompressed domain to avoid the drift effect. Thus, waterscrambling and watermarking processes are applied in a similar embedding system, but in a different manner. For a watermarking scheme, the embedding rule is defined by

$$\forall \vec{d_f} = (d_f^x, d_f^y)^T \in \widetilde{V}_f, \quad \vec{d_f}^{W'} = \vec{d_f} + \widetilde{\Phi}(\alpha, \widetilde{\sigma}_f(W'), K_{\widetilde{\sigma}_f(W')}), \tag{3}$$

in which $\widetilde{\Phi}(\alpha, \widetilde{\sigma}_f(W'), K_{\widetilde{\sigma}_f(W')}) = \alpha \times \widetilde{\Upsilon}(\widetilde{\sigma}_f(W'), K_{\widetilde{\sigma}_f(W')})$, where $\widetilde{\Phi}$ and $\widetilde{\Upsilon}$ are nonreversible functions (e.g., one-way hash functions) ensuring that the mark can only be detected and not extracted, contrary to the waterscrambling scheme. It is important to note that $\widetilde{\sigma}_f(W')$ is not necessarily the same permutation as the one used in the waterscrambling procedure. However, to improve the robustness of this approach, the insertion rule must respect a spatial structure based on the construction of a reference grid $\mathcal{G}$ as illustrated in Figure 3. This rectangular grid is generated in the Cartesian space and is associated to a referential $(O, \vec{i}, \vec{j})$. It represents a block-based partitioning of the image compact support resulting in a set of block elements E, each of size $H \times K$. We denote R_i as the intersection points between blocks that we call here *reference points*.

Each selected motion vector of $\widetilde{V}_f$ is first projected on $\mathcal{G}$ and this projection serves to compute its associated reference point. Figure 3 illustrates this process: the extremity of the projected motion vector $\overrightarrow{OC}$ belongs to a block E of $\mathcal{G}$, from which four intersection points R_1, R_2, R_3, and R_4 can be deduced. The reference point associated to the motion vector is the one located at the smallest distance of the extremity of the vector (according to the L^2 distance). In the example of Figure 3, the reference point of $\vec{d_f}$ is R_1.



(a)

(b)

(c)

(d)

Figure 5: Computation of the watermarked vector.

Then, to embed the watermark, the motion vector is modified (see Figure 4) by constructing in each block element E as a rectangular element of size $h \times k$ (area Z_1), where $h = H - 2 * \delta_1$, $k = K - 2 * \delta_2$, δ_1 and δ_2 are chosen such that Z_1 and Z_2 cover the same area, and $Z_1 \cup Z_2 = E$. Both zones Z_1 and Z_2 drive the mark embedding rule: Z_1 is associated to the bit -1 and Z_2 to the bit $+1$.

Then, if we consider that $\vec{d_f} = \overrightarrow{OC}$ is the vector to be watermarked (see Figure 5) and W_i' is the bit to be inserted, the watermarked vector $\vec{d}_f^{W'}$ is computed as follows:

(i) if $W_i' = +1$ and $\vec{d_f}$ is in the right place (i.e., in the zone Z_2), then $\vec{d}_f^{W'} = \vec{d_f}$; otherwise, a central symmetry of center B must be applied resulting in $\vec{d}_f^{W'} = \overrightarrow{OD}$ (cf. Figure 5b);

(ii) if $W_i' = -1$ and $\vec{d_f}$ is in the right place (i.e., in the zone Z_1), then $\vec{d}_f^{W'} = \vec{d_f}$; otherwise, as the Z_2 area is not compact, three possibilities can appear to compute $\vec{d}_f^{W'}$:

(a) $\vec{d}_f^{W'}$ is given by a central symmetry of center B resulting in $\vec{d}_f^{W'} = \overrightarrow{OD}$ (cf. Figure 5a);

(b) $\vec{d}_f^{W'}$ is given by an axial symmetry parallel to the y-axis and going through B resulting in $\vec{d}_f^{W'} = \overrightarrow{OD}$ (cf. Figure 5c);

(c) $\vec{d}_f^{W'}$ is given by an axial symmetry parallel to the x-axis and going through B resulting in $\vec{d}_f^{W'} = \overrightarrow{OD}$ (cf. Figure 5d).

Note that the case illustrated in Figure 5b (i.e., modification of the motion vector from Z_2 to Z_1) only needs one kind of transformation. Indeed, due to the grid structure and the surface covered by both areas Z_1 and Z_2, a motion vector located within Z_2 will be automatically projected in Z_1 by applying a central symmetry.

In fact, after computing $d_x = C_x - B_x$ and $d_y = C_y - B_y$ (with $B = (B_x, B_y)^T$ and $C = (C_x, C_y)^T$), the symmetry is chosen as follows:

(i) if $d_x \leq \delta_2$ and $d_y \leq \delta_1$, the central symmetry is applied;

(ii) if $d_x \leq \delta_2$, the axial symmetry parallel to the x-axis is applied;

(iii) if $d_y \leq \delta_1$, the axial symmetry parallel to the y-axis is applied.

In this paper, the first attempt to reach the invisibility constraint has conducted us to minimize the distortion applied to the motion vectors, despite it is well known that this rule is not necessarily correlated with the visual aspect of the resulting modified video. To overcome this drawback, we have developed a second approach [40] that consists of choosing the best motion vector in the neighborhood of the original one and which is located in the area corresponding to the bit to be embedded. This last attempt was a significant improvement of the previous one.

```
For $f = 1$–$N$ {//$N$ denotes the video frame number
for $i = 1$–$k_f$ {if $\vec{d}_f^{\,i} \in Z_1$, then $\sigma_f^i(W) = -1$;
        else if $\vec{d}_f^{\,i} \in Z_2$, then $\sigma_f^i(W) = +1$;}}
```

Algorithm 1: Mark detection algorithm.

3.3. *Retrieval Scheme*

Our "dewaterscrambling" procedure is included in an MPEG decompression scheme. The goal of this procedure first consists of extracting the waterscrambled motion vectors, and secondly in dewaterscrambling them to allow the visualization of the original video onto which an invisible watermark has been embedded as detailed in the previous section. As for the embedding process, there are two different possible approaches. The first one uses a syntactic analyzer to extract the marked motion vectors from the MPEG compressed bitstream, to correct them and to re-insert the corrected motion vectors in the MPEG compressed bitstream. The second one consists of directly correcting waterscrambled motion vectors during the decompression scheme. In this case, we must use a module compliant with the standard decompression one.

To dewaterscramble the video, the dewaterscrambling module has to extract the waterscrambling key $K_{\sigma_f(W)}$ in order to initialize the PRNG and the strength α that has been used to waterscramble the video. We recall that the PRNG outputs the k_f indexes k_f^i of the marked motion vectors in V_f. In this way, we can extract the visible watermark from each frame f by applying the inverse function of Φ with

$$\forall \vec{d}_f \in \widetilde{V}_f, \quad \vec{d}_f = \vec{d}_f^{\,W} + \Phi^{-1}\left(\alpha, \sigma_f(W), K_{\sigma_f(W)}\right). \tag{4}$$

For the classical watermarking system, the original video content is not used to detect the watermark presence. As for the waterscrambling system, the key $K_{\widetilde{\sigma}_f(W')}$ is used to initialize the PRNG resulting in the knowledge of the marked motion vectors. The watermark bit inserted in each of the k_f marked vectors $\vec{d}_f^{\,W'}$ can then be detected. For this purpose, we apply the rule illustrated on Algorithm 1.

Once a candidate mark $\widetilde{W}'$ is detected by Algorithm 1, we must decide if it corresponds to the real embedded mark W'. For this purpose, we compute the correlation C_f at frame f between $\widetilde{W}'$ and W' by the following recursion:

$$C_f = \frac{C_{f-1} \times (f-1) + \left(1 - d(\widetilde{W}', W')/N\right)}{f}, \tag{5}$$

where $d(\widetilde{W}', W')$ denotes the Hamming distance between $\widetilde{W}'$ and W' and N is the mark length. If $C_f \geq \theta$, where θ is a predefined correlation threshold, $\widetilde{W}'$ is considered to correspond to W'.

3.4. *Experimental results*

Figure 6 illustrates the video degradation obtained with three different mark strengths α on the Stefan sequence. The first row shows frames of the original sequence ($\alpha = 0$), the second row shows the same frames waterscrambled with $\alpha = 40$, and the last row shows the waterscrambling results with $\alpha = 60$. As expected, the strength α allows the manipulation of the degree of video degradation: the higher the value of α, the higher the degree of video degradation.

We have also conducted some experiments on many sequences to check the robustness of our proposed watermarking scheme. For this purpose, we first performed some of the classical sequence manipulations including Divx[3] lossy compression, blurring with a uniform kernel and rotation. Additionally, we have also tested the robustness of our algorithm on the new codec that appears nowadays, namely H264. The H264 retained model was IBP with quantization steps of 10 and 20. Moreover, all optimization parameters (eighth pixel motion estimation, five reference images, etc.) have been used. The compression ratio used for experiments was 1 : 23 for the Divx codec, 1 : 28 for H264 IBP10, and 1 : 123 for H264 IBP20.

The correlation results (cf. (5)) obtained with these attacks on the Stefan sequence are plotted on Figure 7a. We recall that the correlation level for a frame index f tells us if the mark has been detected in f. In this figure, the correlation threshold θ has been set to $\theta = 0.875$. These results show that the mark is quickly detected, whatever the transform applied onto the sequence. The best result was obtained with the Divx compression that needed only 6 images to detect the mark. The H264 IBP10 compression followed next where the mark was detected after 7 images. Then, the blur attack needed 16 images to detect the mark and the H264 IBP20 model 32 images. Finally, the worst result was obtained with the rotation attack that resulted in a detection at the 82nd frame. We can also remark that the mark was detected in the first frame for the original unattacked sequence. By analyzing these results, we can conclude that our watermark process is particularly robust to all tested attacks and few sequence images are needed to detect the embedded mark.

We then checked the attack consisting in slightly displacing the marked motion vectors. Due to the grid structure used by the embedding rule, when the displaced motion vector remains in the same area (i.e., Z_1 or Z_2), the watermark is right detected. In the other case, the permutation used in the embedding rule ensures that we are able to retrieve the watermark. Indeed, each bit of $\widetilde{\sigma}_i(W')$, never being located in the same place thanks to the statistical accumulation principle, enables the watermark to be retrieved. In both cases, our watermarking scheme proved its robustness against statistical attacks which try to estimate the mark to remove it.

Finally, another kind of attack may consist in recovering the original video content from the waterscrambled one. For this purpose, two kinds of attack can be envisaged:

[3] http://www.divx.com/.

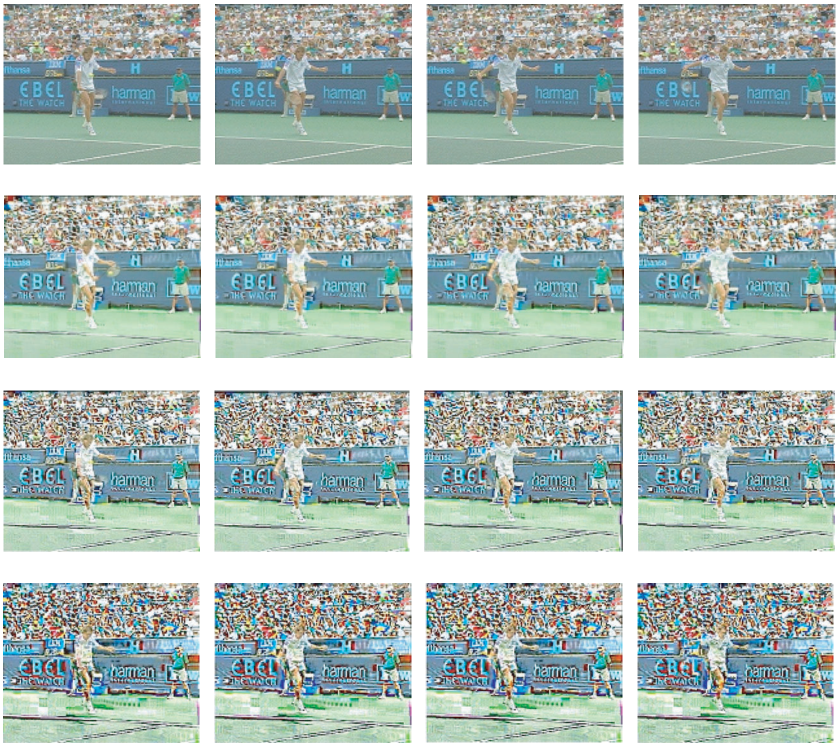


Figure 6: Waterscrambling results on the Stefan sequence according to the strength α. The first row represents original sequence ($\alpha = 0$), the second row represents results with $\alpha = 20$, the third row uses $\alpha = 40$, and the fourth row uses $\alpha = 60$.

(1) from predicted blocks, each being composed of a waterscrambled motion vector and an error block computed from the original motion vector, one can try to correct the waterscrambled motion vectors by retrieving an intra block in the corresponding intra frame that better corresponds to the predicted block. For this purpose, we project back the predicted block into the corresponding intra frame before scanning a search window by looking for an appropriate intra block that maximizes the PSNR with the current predicted block;

(2) from intra blocks located in P or B frames (which are not waterscrambled since not being predicted), one can try to retrieve the original associated motion vectors of neighboring predicted blocks by increasing the block smoothness between the unwaterscrambled intra blocks and the waterscrambled predicted blocks.

The implementation of these two attacks led to the following conclusions:

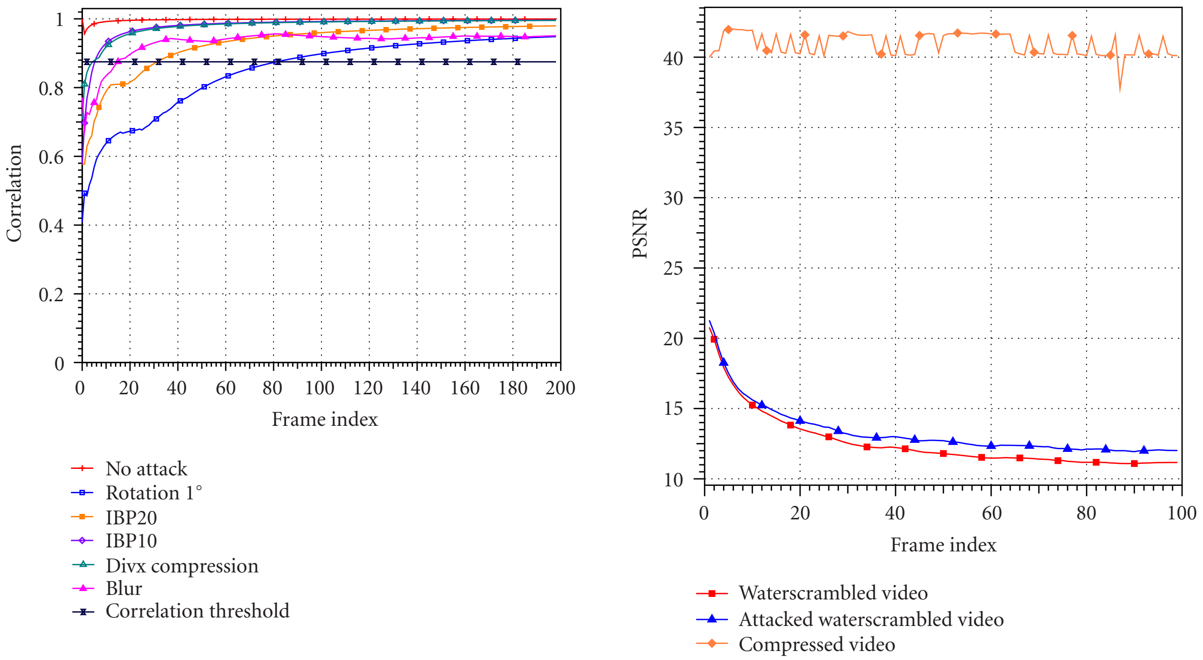
(i) the first attack generates many block artifacts on the block boundaries. These artifacts can be explained by the fact the attacker only gets the error block to retrieve the original intra block corresponding to the predicted block. Therefore, he can only find an approximation of the original block and an approximation of the original motion vector. This approximation error causes highly visible artifacts that are drastically increased with time, being due to the drift effect;

(ii) the second attack has two drawbacks. The first one is explained by the fact that there are generally few intra blocks in a predicted frame. Thus, we are only able to approximatively correct the motion vectors of predicted blocks located in the immediate neighborhood of intra blocks, but we are unable to ensure the correction of motion vectors when we walk far from the initial intra block because the approximation error grows in the spatial domain each time we consider a new predicted block to process. The other drawback concerns the accumulation of the same approximation error in the temporal axis generating a drift effect.

During our experiments, the first attack gave better results than the second one and to give an idea about the visual effect of this attack, we have plotted PSNR curves in Figure 7b. This figure shows the PSNR of the compressed video sequence, the unattacked waterscrambled sequence, and the attacked waterscrambled sequence. As we can see, the attack allows to slightly increase the PSNR, which remains nevertheless below the PSNR of the original sequence, proving that this attack is not effective.

4. CONCLUSION AND PERSPECTIVES

In this paper, we have presented an alternative video protection model called *waterscrambling*. This new model has been motivated by the proposal of a new solution able to show a degraded form of content in order to provoke an impulsive



(a)

(b)

Figure 7: (a) Correlation results on the Stefan sequence and (b) PSNR gain offered by the proposed attack on the same sequence.

buying action in some applications such as e-commerce. The developed solution consists of disturbing the video sequence motion vectors in the compressed domain. Then, a complementary watermarking solution based on a similar approach has been added to our process in order to prevent illegal copy. We have created a new rule of embedding process by adding the invisible mark to previously waterscrambled motion vectors in order to maintain a real-time process. Therefore, while the motion vectors are visually disturbed for the waterscrambling scheme, we add only a small perturbation on them for the watermarking approach, making the unwaterscrambled content similar to the original. Moreover, while the waterscrambling scheme uses a reversible function during the mark embedding, the watermarking scheme uses a nonreversible function. The results reached by these two processes show their respective potential. Effectively, waterscrambling appears to be a good intermediate between the scrambling and watermarking process. Although our watermarking process first shows good results, it may not be robust against format changes, like scaling or shrinking. But our first goal was to resist compression and filter attacks. Now, future works will be concentrated on the improvement of our watermarking scheme in order to increase its robustness against other classes of attacks (particularly geometrical attacks). Additionally, the rule used to scramble the motion vectors has to be improved to keep their statistical distribution and consequently to keep the same level of compression. Finally, a fully compressed domain system has to be developed by taking care of the constraints brought by this domain on the watermarking scheme (e.g., drift effect).

REFERENCES

[1] Commission of the European Communities, “Digital rights: background, systems, assessment,” Comission Staff Working Paper SEC(2002)197, Brussels, February 2002.

[2] B. M. Macq and J.-J. Quisquater, “Cryptology for digital TV broadcasting,” *Proceedings of the IEEE*, vol. 83, no. 6, pp. 944–957, 1995.

[3] J. A. Bloom, I. J. Cox, T. Kalker, J.-P. M. G. Linnartz, M. L. Miller, and C. B. S. Traw, “Copy protection for DVD video,” *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1267–1276, 1999.

[4] F. A. Stevenson, “Cryptanalysis of contents scrambling system,” online paper, http://www-2.cs.cmu.edu/~dst/decss/frankstevenson/analysis/html, November 1999.

[5] B. Schneier, *Applied Cryptography*, John Wiley & Sons, New York, NY, USA, 2nd edition, 1996.

[6] A. Piva, F. Bartolini, and M. Barni, “Managing copyright in open networks,” *IEEE Internet Computing*, vol. 6, no. 3, pp. 18–26, 2002.

[7] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, “Information hiding—a survey,” *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, 1999, Special issue on protection of multimedia content.

[8] I. J. Cox, M. L. Miller, and J. A. Bloom, “Watermarking applications and their properties,” in *Proc. IEEE International Conference on Information Technology: Coding and Computing*, pp. 6–10, Las Vegas, Nev, USA, March 2000.

[9] S. Katzenbeisser and F. A. P. Petitcolas, Eds., *Information Hiding. Techniques for Steganography and Digital Watermarking*, Artech House, London, UK, 2000.

[10] G. C. Langelaar, I. Setyawan, and R. L. Lagendijk, “Watermarking digital image and video data. A state-of-the-art overview,” *IEEE Signal Processing Magazine*, vol. 17, no. 5, pp. 20–46, 2000.

[11] N. Nikolaidis and I. Pitas, "Digital image watermarking: an overview," in *Proc. IEEE International Conference on Multimedia Computing and Systems*, vol. 1, pp. 1–6, Florence, Italy, June 1999.

[12] J. J. K. O'Ruanaidh, W. J. Dowling, and F. M. Boland, "Watermarking digital images for copyright protection," *IEE Proceedings on Vision, Image and Signal Processing*, vol. 143, no. 4, pp. 250–256, 1996.

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

[14] F. Hartung and G. Girod, "Watermarking of uncompressed and compressed video," *Signal Processing*, vol. 66, no. 3, pp. 283–301, 1998.

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

[16] J. Dittman, M. Stabenau, and R. Steinmetz, "Robust MPEG video watermarking technologies," in *Proc. 6th ACM International Conference on Multimedia*, pp. 71–80, Bristol, England, September 1998.

[17] J. Fridrich, "Methods for data hiding," Tech. Rep., Center for Intelligent Systems & Department of Systems Science and Industrial Engineering, Suny Binghamton, Binghamton, NY, USA, 1997.

[18] R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp, "Perceptual watermarks for digital images and video," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1108–1126, 1999.

[19] T. Kalker, G. Depovere, J. Haitsma, and M. Maes, "Video watermarking system for broadcast monitoring," in *Proc. SPIE Conference on Security and Watermarking of Multimedia Contents*, vol. 3657, pp. 103–112, San Jose, Calif, USA, January 1999.

[20] R. Maandonks, T. Kalker, and J. Haitsma, "Video watermarking content protection technology," in *Audio Video & Broadcasting Studio Systems*, pp. 26–30, November–December 2001.

[21] E. Koch and J. Zhao, "Embedding robust label into images for copyright protection," in *Proc. International Congress on Intellectual Property Rights for Specialized Information, Knowledge and New Technologies (KnowRight '95)*, pp. 242–251, Vienna, Austria, August 1995.

[22] S. Baudry, P. Nguyen, and H. Maitre, "Use of synchronisation patterns to estimate geometric distorsions in digital watermarking," in *Proc. of the 11th European Signal Processing Conference (Eusipco '02)*, pp. 709–712, Toulouse, France, September 2002.

[23] G. Doërr and J.-L. Dugelay, "A guide tour of video watermarking," *Signal Processing: Image Communication*, vol. 18, no. 4, pp. 263–282, 2003.

[24] F. Deguillaume, G. Csurka, J. J. K. O'Ruanaidh, and T. Pun, "Robust 3D DFT video watermarking," in *Proc. of the 11th Annual Symp. of Electronic Imaging '99: SPIE Security and Watermarking of Multimedia Contents*, vol. 3657, pp. 113–124, San Jose, Calif, USA, January 1999.

[25] A. H. Tewfik, M. D. Swanson, and B. Zhu, "Multiresolution scene-based video watermarking using perceptual models," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 540–550, 1998.

[26] K. F. Jordan, M. Kutter, and T. Ebrahimi, *Proposal of a watermarking technique for hiding/retrieving data in compressed and decompressed video*, ISO/IEC/JTC1/SC29/WG11/MPEG97/M2281, July 1997.

[27] J. Zhang, J. Li, and L. Zhang, "Video watermark technique in motion vector," in *Proc. 14th Brazilian Symp. on Computer Graphics and Image Processing (SIBGRAPI '01)*, pp. 179–182, Florianopolis, Brazil, October 2001.

[28] R. Lancini, F. Mapelli, and S. Tubaro, "A robust video watermarking technique in the spatial domain," in *Proc. 4th EURASIP-IEEE Symposium on Video/Image Processing and Multimedia Communications*, pp. 251–256, Zadar, Croatia, June 2002.

[29] T. Kalker and J. Haitsma, "Efficient detection of a spatial spread-spectrum watermark in MPEG video streams," in *Proc. IEEE International Conference on Image Processing (ICIP '00)*, vol. 1, pp. 434–437, Vancouver, BC, Canada, September 2000.

[30] J. Haitsma and T. Kalker, "A watermarking scheme for digital cinema," in *Proc. IEEE International Conference on Image Processing (ICIP '01)*, vol. 2, pp. 487–489, Thessaloniki, Greece, October 2001.

[31] P. Bas and B. Macq, "A new video-object watermarking scheme robust to object manipulation," in *Proc. IEEE International Conference on Image Processing (ICIP '01)*, vol. 2, pp. 526–529, Thessaloniki, Greece, October 2001.

[32] A. Piva, R. Caldelli, and A. De Rosa, "A DWT-based object watermarking system for MPEG-4 video streams," in *Proc. IEEE International Conference on Image Processing (ICIP '00)*, vol. 3, pp. 5–8, Vancouver, BC, Canada, September 2000.

[33] W. Macy, R. Liu, M. Holliman, B.-L. Yeo, and M. Yeung, "Content protection and delivery system for the internet," in *Proc. IS&Ts PICS Conference*, pp. 279–284, Savannah, Ga, USA, April 1999.

[34] F. Bao, "Multimedia content protection by cryptography and watermarking in tamper-resistant hardware," in *Proc. ACM Multimedia Conference*, pp. 139–142, Los Angeles, Calif, USA, October-November 2000.

[35] H. Cheng and X. Li, "Partial encryption of compressed images and videos," *IEEE Transactions on Signal Processing*, vol. 48, no. 8, pp. 2439–2451, 2000.

[36] W. Zeng and S. Lei, "Efficient frequency domain selective scrambling of digital video," *IEEE Transactions on Multimedia*, vol. 5, no. 1, pp. 118–129, 2003.

[37] S. Roche, J.-L. Dugelay, and R. Molva, "Multi-resolution access control algorithm based on fractal coding," in *Proc. IEEE International Conference on Image Processing (ICIP '96)*, vol. 3, pp. 235–238, Lausanne, Switzerland, September 1996.

[38] J. Kelsey, B. Schneier, and N. Ferguson, "Yarrow-160: notes on the design and analysis of the yarrow cryptographic pseudorandom number generator," in *6th Annual International Workshop on Selected Areas in Cryptography (SAC '99)*, vol. 1758 of *Lecture Notes in Computer Science*, p. 13, Springer Verlag, Kingston, Ontario, Canada, August 1999.

[39] G. Girod and F. Hartung, "Watermarking method and apparatus for compressed digital video," US Patent 5,809,139, September 1998.

[40] Y. Bodo and N. Laurent, "Une technique de watermarking vidéo basée sur le marquage hiérarchique de vecteurs de mouvement par sélection adaptative des zones d'insertion," French patent FR 03 04590, France Telecom, April 2003.

Yann Bodo received the Engineer Degree in applied mathematics from the National Institute of Applied Science (INSA), Rouen, France, in 2000. He is currently pursuing the Ph.D. degree in image processing, ENST, Paris. In 2000, he joined France Telecom R&D, Rennes, as a member of the technical staff researching issues in the watermarking of video data.

Nathalie Laurent was awarded a Ph.D. degree in applied mathematics in 1997 from the University of Bordeaux, France. Since 1997, she works in the Human Interaction Division of France Telecom R&D at Rennes, France. Her research interests include video coding, copyright protection for multimedia contents, image retrieval, and face detection and recognition.

Christophe Laurent received the Ph.D. degree in computer science from the University of Bordeaux, France, in 1998. During his Ph.D., he worked on parallel computing for image processing. In 1998, he joined Thomson Multimedia where he worked as a Senior Researcher on security of information systems. During this period, he was specialized on network security and security for e-commerce applications. Since 2001, he works as a researcher at France Telecom R&D where his research interests include image indexing, pattern recognition, and face processing.

Jean-Luc Dugelay received the Ph.D. degree in computer science in 1992 from the University of Rennes. Doctoral research was carried out, from 1989 to 1992, at the France Telecom Research Laboratory in Rennes (formerly CNET - CCETT). He then joined the Institut Eurécom (Sophia Antipolis), where he is currently a Professor in the Department of Multimedia Communications. His research interests are in the area of multimedia signal processing and communications; including security imaging (i.e., watermarking and biometrics), image/video coding, facial image analysis, virtual imaging, face cloning, and talking heads. He is an author or coauthor of more then 65 publications that have appeared as journal papers or proceeding articles, 3 book chapters, and 3 international patents. He gave several tutorials on digital watermarking and image compression at major conferences. He has been an invited speaker and/or member of the program committee of several scientific conferences and workshops. Jean-Luc Dugelay is an Associate Editor for several major international journals, and an active member of the IEEE Signal Processing Society.