A Bayesian Approach To Affine Transformation Resistant Image and Video Watermarking

Gabriella Csurka[1], Frédéric Deguillaume[1], Joseph J. K. Ó Ruanaidh[1,2] and Thierry Pun[1]

[1] CUI, University of Geneva, 24 rue Général Dufour, CH 1211 Geneva, Switzerland
{Gabriela.Csurka,Frederic.Deguillaume,Thierry.Pun}@cui.unige.ch
http://cuiwww.unige.ch/~vision
[2] Siemens Corporate Research, 755 College Road East, Princeton, NJ 08540, US
oruanaidh@scr.siemens.com

Abstract. This paper proposes a new approach for assessing the presence of a digital watermark in images and videos. This approach relies on a Bayesian formulation that allows to compute the probability that a watermark was generated using a given key. The watermarking itself relies on the discrete Fourier transform (DFT) of the image, of video frames or of three dimensional chunks of video scene.
Two kinds of information are hidden in the digital data: a watermark and a template. Both are encoded using an owner key to ensure the system security and are embedded in the 2D or 3D DFT magnitude of the image or video. The watermark is encoded in the form of a spread spectrum signal. The template is a key based grid and is used to detect and invert the effect of an affine geometric transformation (rotations, scaling and/or aspect ratio change) or in the case of the video frame-rate changes, aspect-ratio modification and rescaling of frames.
It is shown that, for the spread spectrum signal described in the paper, it is possible to positively detect the presence of a watermark without necessarily decoding it, given only the key used to generate it. To do that, the Bayesian approach allows the deduction of the relative log-probability of the presence of a watermark in an image given only the key. All that is required for positive watermark verification is the key. This approach also allows to infer the number of bits contained in the watermark.
The performance of the presented techniques and the usefulness of the relative log-probability are shown through various applications and experimental results.

1 Introduction

The popularity of the World Wide Web has clearly demonstrated the commercial potential of the digital multimedia market and consumers are investing heavily in digital audio, image and video recorders and players. Unfortunately, digital networks and multimedia also afford virtually unprecedented opportunities to

pirate copyrighted material. Digital storage and transmission make it trivial to quickly and inexpensively construct exact copies. Therefore, digital image and video watermarking and copyright protection has recently become a very active area of research.

The main idea of the method presented here is to use a *Bayesian approach* to image and video watermark detection, in order to compute the probability that a watermark was generated using a given key (section 5). All that is required for positive watermark verification is the key. This technique works even when the binary message cannot be decoded; this is interesting, because it suggests that the watermark algorithm can be used either to extract binary messages, or for binding a watermark to the key, i.e. it is not necessary to decode the watermark to verify the ownership.

The watermark used in this paper is a Gold Code based spread spectrum sequence (see section 3) which encodes the message, containing information such as owner identification, a serial number, type of content, etc., or alternatively a hash number to a table that contains these elements. The major advantage of a spread spectrum system is that it transforms the narrow band data sequence into a noise-like wide-band signal, using pseudo-random sequences that are difficult to detect and extract.

The watermark, in the form of a spread spectrum sequence, is embedded in a mid-frequency range of the discrete Fourier transform (DFT) domain of the image or video as a compromise between the visibility of the watermark and robustness to lossy compression and other attacks (section 3). Due to the properties of the Fourier transform and spread spectrum techniques this method resists to spatial and/or temporal shifts, to simple filtering, noise adding, JPEG respectively MPEG compression. However, if the modification suffered by the image or video are geometric transformations such as rotation, scaling, cropping, aspect ratio change respectively changes of frame-rate the positions where the mark was embedded also change. In order to be able to synchronize and to decode the message in these cases, a template is used to detect and invert the effect of a *general affine geometric transformation* undergone by the image, or a shift in the log-log-log map[1] in the 3d DFT of the video prior to extracting the mark (section 4).

Several applications are presented in the second part of the paper, which show the usefulness of the proposed Bayesian approach (section 6).

2 Background

Most of the current watermarking techniques for digital images, inspired by methods of image and video coding and compression, work in the transformed frequency domain. The reason is that data loss usually occurs among the high

[1] The reason not to search for affine transform in the case of videos is that, from one hand the 3D linear transform depends on more parameters and the template search has a high computational cost and on the other hand it unlikely that video frames be rotated or transformed by a random linear transform.

frequency components. Hence the watermark has to be placed in the significant frequency components of the Discrete Cosine Transform (DCT) [1] or Discrete Fourier Transform (DFT) [8,9] of the image. Some other approaches have been proposed that rather embed the mark in the image in the spatial domain [7,5]. Geometric distortions are determined manually [1], using an empirical search [5,7], using template-grid and automatic template matching [9] or working in a geometric distortion invariant space [8].

In the case of video watermarking the challenge is to mark a group of images which are strongly intercorrelated and often manipulated in a compressed form, e.g. MPEG. A first group of video watermarking methods therefore directly operate on MPEG data to avoid full decompression [4]. Other approaches, less dependent on the type of video compression and more resistant to rescaling, change of frame-rate and re-compression in a different format, mark the uncompressed video sequence. Each video frame is then marked individually as a still image or the video is considered as a three-dimensional signal and the mark is embedded into the 3D DFT domain [2]. In another approach, Swanson *et al.* [13] make use of a temporal wavelet transform of the video; 2D wavelet frames are then marked instead of the original video frames.

One of the key element to make a watermark robust is to embed it in the perceptually significant components of the image taking account of the behavior of the human visual system (HVS) [1]. Another key element is the use of spread spectrum techniques to encode the information before embedding it in the images [14,1]. A good spread spectrum sequence is one which combines desirable statistical properties such as uniformly low cross correlation with cryptographic security. It was shown in [11] that m-sequences and Gold Code based spread spectrum sequences perfectly fulfil these requirements.

3 The Watermark

The approach presented here uses spread spectrum to encode the message, that can contain information such as the owner of the image, a serial number and perhaps flags which indicate the type of content e.g. religion, pornography, or politics, or alternatively a hash number to a table that contains these informations. System security is based on proprietary knowledge of the keys (or the seeds for pseudo-random generators) which are required to embed, extract or remove an image watermark.

Let the message be represented in binary form as $\widehat{\boldsymbol{b}} = (\hat{b}_1, \hat{b}_2, ...\hat{b}_M)^\top$ where $\hat{b}_i \in \{0,1\}$ and M is the number of bits in the message to be encoded. The binary form of the message $\widehat{\boldsymbol{b}}$ is then transformed to obtain the vector $\boldsymbol{b} = (b_1, b_2, ...b_M)^\top$, with $b_i \in \{1,-1\}$ by exploiting the basic isomorphism between the group[2] ($\oplus$,{0,1}) and the group (*,{1,-1}). The mapping $1 \rightarrow -1$ and $0 \rightarrow 1$ is an extremely important step because it essentially enables us to replace the exclusive-OR operator used in finite field algebra with multiplication.

[2] The bit addition modulo 2, $\oplus$ is equivalent to exclusive-OR.

Defining a set of random sequences $\boldsymbol{v}_i$ each corresponding to a bit b_i, the encoded message can be obtained by:

$$\boldsymbol{w} = \sum_{i=1}^{M} b_i \boldsymbol{v}_i = \mathbf{G}\boldsymbol{b} \tag{1}$$

where $\boldsymbol{b}$ is a $M \times 1$ vector of bits (in ± 1 form), $\boldsymbol{w}$ is a $N \times 1$ vector and $\mathbf{G}$ in $N \times M$ matrix such that the i^{th} column is a pseudo-random vector $\boldsymbol{v}_i$.

Clearly, the effectiveness of this scheme depends on the specific choice for the random vectors $\boldsymbol{v}_i$. It is shown in [11], how pseudo-random sequences can be used to spread the signal spectrum. In order to despread the signal these pseudo-random sequences need to have good randomness properties, long periods and different sequences must be well separated in term of correlation. In other words, the specific choice of method for generating the pseudo-random sequence has direct implications to the reliability and security of the embedded mark.

Maximum length sequences or simply m-sequences perfectly fulfil these requirements. They are the largest codes that can be generated by a shift register of a given length. They exist for all integer values n with period $N = 2^n - 1$ and can be easily generated by proper connections of feedback paths in an n-stage shift register circuit [11]. The auto-correlation function and spectral distribution resemble that of white Gaussian noise. Cross-correlation between shifted versions of m-sequences are equal to -1, whereas autocorrelations are equal to N. Therefore, one alternative to choose $\boldsymbol{v}_i$ to encode the message $\boldsymbol{b}$ is to consider an m-sequence $\boldsymbol{v}_1$ and for each i, $\boldsymbol{v}_{i+1}$ results from $\boldsymbol{v}_i$ by a circular shift of length 1 (each element of $\boldsymbol{v}_i$ is shifted to right and the last element becomes the first).

An other alternative for $\boldsymbol{v}_i$ is to use Gold Codes [11,3]. A family of Gold Codes is obtained using an m-sequence $\boldsymbol{v}_1$ and a q-decimation of it. The decimation $\boldsymbol{v}'_1$ is obtained by sampling every q^{th} element of $\boldsymbol{v}_1$. Note that $\boldsymbol{v}'_1$ has period N if and only if gcd(N,q)=1, where "gcd" denotes the greatest common divisor. Each element $\boldsymbol{v}_{i+1}$ of the family can then be obtained as follows: take $\boldsymbol{v}'_1$ (of length N), circular shift it by i and multiply element by element with the vector $\boldsymbol{v}_1$.

Gold sequences have the advantage that for a given register length n there are more choices for the "key" than with shifted m-sequences. Indeed, for a register length n, there are $2^n - 1$ possible m-sequences ($2^n - 1$ possible seed as initial element in the register) but for Gold Codes there is in addition the possibility to choose a different decimation q. In addition, it is known that Gold sequences have better cross correlation properties if only part of the sequence is used. This could have implications if the watermark is partially destroyed by image cropping or filtering.

The watermark, in the form of a spread spectrum sequence, is embedded in a mid-frequency range of the discrete Fourier transform (DFT) domain of the image or video. The upper and the lower frequency bound are fixed in advance as a compromise between the visibility of the watermark (low frequencies contain the most of the image information) and robustness to lossy compression (which remove high frequencies). Note that for videos the mark can be embedded in

the 2D DFT of each frame as for still images or into the magnitude of the 3D DFT of video chunks [2]. In the case of the 3D DFT the third dimension corresponding to the temporal frequencies also has to be taken into account. These frequencies have the following properties. Null or low temporal frequencies are linked to static components in the input scene, while higher frequencies are related to moving objects and varying areas. Therefore again, due this time to a compromise between the static and moving components, a mid range is considered.

The spread spectrum message $\boldsymbol{w}$ is added only to the magnitudes of the DFT and the phase is left unaltered. The strength of the watermark, corresponding to a modulation factor ($s_w \boldsymbol{w} \rightarrow \boldsymbol{w}$) can be set either interactively or adaptively as a function of the average and standard deviation of the DFT components of the chosen frequency range. Note, that $\boldsymbol{w}$ contains positive and negative values (it is easy to see from (1) that $w_i \in \{-M, M\}$). To have a better security, for each w_i a pair of location $\{(u_i^1, v_i^1), (u_i^2, v_i^2)\}$ in the specified frequency band of the magnitude domain is chosen pseudo-randomly in function of the owner key. Then, if $w_i > 0$, it is added to the magnitude at the position (u_i^1, v_i^1), otherwise to the magnitude at the position (u_i^2, v_i^2). To be more robust, instead of simply adding the values of the spread spectrum, one can modify the magnitude values at (u_i^1, v_i^1) and (u_i^2, v_i^2) in such a way that their difference becomes equal to w_i. Finally, from the marked spectrum and the unchanged phase the DFT is inverted (inverse Fourier transform) yielding the watermarked image.

In order to extract the watermark, the DFT of the marked image or video is considered. As the pair of positions of the magnitude components modified is known (it depends on the owner key), there is no need of the original image to extract the watermark. The difference between the pair-wise coefficients allows to obtain a spread spectrum signal $\boldsymbol{w'} = \boldsymbol{w} + \boldsymbol{e}$, where $\boldsymbol{w}$ is the embedded watermark and $\boldsymbol{e}$ is an additive error. In order to decode the message from $\boldsymbol{w'}$, for each i the dot product ("cross-correlation") between $\boldsymbol{v}_i$ and $\boldsymbol{w'} = \sum_{i=1}^{M} b_i \boldsymbol{v}_i + \boldsymbol{e}$ is performed:

$$B'_j = < \boldsymbol{w'}, \boldsymbol{v}_j > = \sum_{i=1}^{M} b_i < \boldsymbol{v}_i, \boldsymbol{v}_j > + < \boldsymbol{e}, \boldsymbol{v}_j > \tag{2}$$

It can be shown that for m-sequences and Gold Codes $< \boldsymbol{v}_i, \boldsymbol{v}_j > = -1$ for $i \neq j$ and $< \boldsymbol{v}_i, \boldsymbol{v}_i > = N$. Replacing them in (2), gives $B'_j = b_i N - (M-1) + < \boldsymbol{e}, \boldsymbol{v}_j >$. Generally $M \ll N$. Moreover the distribution of $\boldsymbol{e}$ can be approximated by a normal distribution with zero mean, so $< \boldsymbol{e}, \boldsymbol{v}_j >$ is negligible comparing to N. Therefore, each embedded information bit b_j can be retrieved as follows:

$$b'_j = sign(B'_j) = sign(b_j) = b_j$$

4 Affine transformation resistant watermarking

The watermarking method described in the previous section presents an inherent invariance to spatial shifts for images and videos and/or temporal shift for

videos due to the basic properties of the Fourier transform. It also resists to simple filtering, noise adding, JPEG respectively MPEG compression because the spread-spectrum sequences are very robust to noise or partial cancelation. However, if the modification suffered by the image or video are geometric transformations such as affine geometric transformation (rotations, scaling and/or aspect ratio change), cropping[3] respectively changes of frame-rate the positions where the mark was embedded also change. In order to be able to synchronize and to decode the message in these cases, a *template* is used to detect and invert the geometric transformation undergone by the image or video prior to extracting the mark.

The template is defined as a sparse set of positions in the 2D or 3D DFT. These positions are owner key dependent and therefore it is possible to create a reference template during the extraction. The magnitude values at these positions are modified to become local maxima (peaks) in the DFT magnitude domain in order to facilitate the template search process.

4.1 Template Search in 2D DFT. First, all local maxima (peaks) in the magnitude of the 2D DFT are extracted. The geometric transformation is then estimated by a point matching algorithm between the extracted peaks and the reference template points as follows. It is assumed that the more general global transformation applied to an image is an affine transformation (which is a linear transform defined by a 2×2 matrix $\mathbf{T}$ plus a translation). Due to the properties of the Fourier transform, the magnitude is invariant to the translation, and therefore the corresponding transformation in the DFT magnitude depends only on four parameters (the coefficients of $\mathbf{T}^{-\top}$). Consequently, it is sufficient to have two pairs of matches between the local peaks and the template points to be able to estimate the geometric transformation. An exhaustive search is performed considering each pair of peaks with each pair of template point. The corresponding linear transformation is applied to all the other template points and the transformation for which there is a maximum number of match between the peaks and the transformed template points is retained. In order to optimize the exhaustive search, only those pairs of matches are considered that satisfy some predefined constraints. They concern the distance between the pairs of points, the angle defined by the pair of point and the origin of the DFT (0 frequency) and the order of the points as a function of the distance from the origin, before (reference template points) and after the transformation (the considered pair of peaks). These constraints are based on the assumption that the geometric distortion of the image cannot be very important. Finally, to have a more accurate linear transform, the linear transformation is reestimated using a least median square technique on the set of found matches.

4.2 Template Search in 3D DFT. One alternative is to generalize the above technique for the 3D DFT. However, the 3D linear transform depends on 9 pa-

[3] Note that cropping in direct domain corresponds to a horizontal and/or vertical scaling in the frequency domain, and therefore it is a particular case of a linear transform in the DFT.

rameters (three matches) and therefore the template search would have a high computational cost. Furthermore, it is unlikely that video frames be rotated or transformed by a random linear transform. Generally, the changes suffered by a video are frame cropping, frame scaling and/or changes of aspect ratio, or changes of frame-rate. Therefore, instead of a generalization of the above technique, a log-log-log map of the 3D DFT is used which allows to find independent scalings along the three axes corresponding to a aspect-ratio change and a frame-rate change. The log-log-log is a bijective function (if the origin $(0,0,0)$ was extracted) and it converts the DFT (k_x, k_y, k_z) space to a (μ_x, μ_y, μ_z) logarithmic space as follows:

$$\mu_t = sign(k_t) \cdot ln(|k_t|), \quad \text{where} \quad t \in \{x, y, z\} \tag{3}$$

From the signed shifts $\triangle\mu_x$, $\triangle\mu_y$ and $\triangle\mu_z$ in the log-log-log space obtained by the template matching the rescaling factors s_x, s_y, s_z of the 3D DFT are $s_x = e^{\triangle\mu_x}$, $s_y = e^{\triangle\mu_y}$ and $s_z = e^{\triangle\mu_z}$.

Consequently, in the log-log-log map the scalings along the three axes are transformed to a simple 3D shift. This means that in order to retrieve the transformation a cross-correlation step needs to be applied between the reference template and the mapped DFT. However the cross-correlation can be computationally costly, and can be replaced by a point matching techniques between the local peaks and template points. As the 3D shift is defined by 3 parameters, it is sufficient to have a match between a local peaks and a template points to be able to estimate it. An exhaustive search is therefore performed considering each peak with each template point, and the corresponding shift is applied to all the other template points. The shift for which there is a maximum number of matches between the peaks and the transformed template points is retained.

5 The Bayesian Approach for Assessing the Watermark Presence

Consider a binary string extracted $\boldsymbol{b}'$ from an image, for which almost all bits agree with a known binary sequence message $\boldsymbol{b}$ which may have been embedded in the image. Generally, one can expect that about 50% of the bits of a random sequence will agree with the watermark. The question that can be asked is: What is the probability that the almost perfect agreement occurred at random? In order to answer this question, consider the probability that a random sequence will have a certain number of bits in common with a given sequence. It is easy to show that this probability is given by the Bernoulli distribution: $p(i) = C_N^i / 2^N$, where N is the number of bits in both messages and i is the number of bits found to be in common.

The implications of this result are quite far reaching. If one decodes a 100 bit watermark and finds that 80% of the bits are "correct" then one can be fairly sure that the watermark was indeed found. This is because the probability of getting 80% or more bits correct is at random is 2.17×10^{-9}. This is the

probability of a false alarm – where one would say that a watermark is present when in fact there is none.

However, generally the original message is generally not known, and a bit by bit comparison between the extracted message (which can be corrupted) and the original one is not possible. Being able to detect a watermark without necessarily being able to decode it, can be therefore highly useful since it can help to prove ownership (the owner of the given key) in the case when due to noise the message can only partially be decoded. One expects that watermark detection will always be more robust than watermark decoding because in detection one is essentially transmitting a single bit of information which is to say whether a watermark is present or not.

In which follows, a new approach is proposed that computes the probability that the watermark was generated using a given key. The basic idea is as follows. A watermark $\boldsymbol{w}$ is embedded in the DFT domain of the image or video. The watermark is a m-sequence or Gold Code based spread spectrum sequence $\boldsymbol{w} = \mathbf{G}\boldsymbol{b}$ (see (1)) and the extracted watermark can generally be estimated by $\boldsymbol{w}' = \mathbf{G}\boldsymbol{b} + \boldsymbol{e}$, where $\boldsymbol{e}$ is an additive noise vector corrupting the watermark.

Assuming that the noise is approximately Gaussian distributed and applying the Bayesian approach described in [12], the *probability* that a spread spectrum signal $\boldsymbol{w}'$ extracted from the image I' contains a message of length M encoded with the key K can be computed as follows:

$$p\left(K, M \mid \boldsymbol{w}', I'\right) \propto \frac{\pi^{-N/2}\, \Gamma\left(\frac{M}{2}\right) \Gamma\left(\frac{N-M}{2}\right) \left|\mathbf{G}^\top \mathbf{G}\right|^{-1/2}}{4\, R_\delta\, R_\sigma \left(\boldsymbol{b}'^\top \boldsymbol{b}'\right)^{M/2} \left(\boldsymbol{w}'^\top \boldsymbol{w}' - \boldsymbol{f}^\top \boldsymbol{f}\right)^{(N-M)/2}} \tag{4}$$

where $\boldsymbol{b}' = \left(\mathbf{G}^\top \mathbf{G}\right)^{-1} \mathbf{G}^\top \boldsymbol{w}'$ and $\boldsymbol{f} = \mathbf{G}^\top \boldsymbol{b}$, are the least squares estimate for the bit sequence and the fit of the estimated watermark respectively. R_σ and R_δ are irrelevant constants introduced as normalization factors.

For m-sequences and Gold sequences, the $M \times M$ matrix $\mathbf{A} = \mathbf{G}^\top \mathbf{G}$ has the following form: $\mathbf{A}(i,j) = -1$ for $i \neq j$ and $\mathbf{A}(i,i) = N$. The two terms in expression (4) which require the most computation are $\mathbf{B} = (\mathbf{G}^\top \mathbf{G})^{-1}$ and $\left|\mathbf{G}^\top \mathbf{G}\right|$. However, both the determinant and the inverse can be computed in closed form:

$$\left|\mathbf{G}^\top \mathbf{G}\right| = (N+M-1)(N+1)^{M-1} \quad \text{and} \quad \mathbf{B}(i,j) = \begin{cases} \frac{2^N - M + 1}{2^N\,(2^N - M)} & \text{if } i = j \\ \frac{1}{2^N\,(2^N - M)} & \text{if } i \neq j \end{cases}$$

In a similar way, the *probability that no message* (a message of length 0) was embedded in the image with a given key K is:

$$p(K,\, 0 \mid \boldsymbol{w}',\, I') \propto \frac{\pi^{-N/2} \Gamma\left(\frac{N}{2}\right)}{2R_\sigma (\boldsymbol{w}'^\top \boldsymbol{w}')^{N/2}} \tag{5}$$

Finally, to decide if a given key was used or not to generate a watermark, the *relative log-probability*:

$$P_r = log\Big(\frac{p(K,\, M \mid \boldsymbol{w}',\, I')}{p(K,\, 0 \mid \boldsymbol{w}',\, I')}\Big) \tag{6}$$

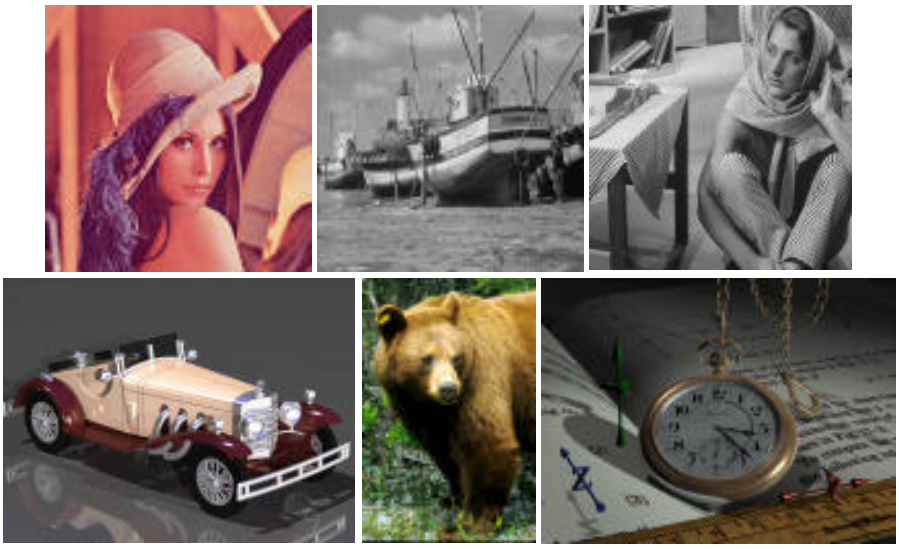

Fig. 1. Six marked images used to test. Up: Lena (color, 512 × 512, PSNR=37.2), Fishingboat (gray-scale, 512 × 512, PSNR=39.5) and Barbara (gray-scale, 256 × 256, PSNR=32.5), Down: Benz (color, 640 × 480, PSNR=42.2), Bear (color, 394 × 600, PSNR=39.6) and Watch (color, 1024 × 768, PSNR=44.1).

is compared to 0. If $P_r > 0$ a message was embedded with the key K, otherwise no message was embedded with K. To exclude false alarms, a threshold $t > 0$ can be chosen by accepting occasional false rejections (noisy cases).

6 Applications of the Bayesian approach

In the following, we show some experimental results and some concrete applications of this Bayesian approach. In our experiments, we used several real images and videos. In each cases we encoded the message "CUI_Tests" (72 bit) with the key (seed) 180599. Figure 1 shows the marked images[4].

6.1 Searching for the message length. We are interested here in determining the message length M using the Bayesian approach. Therefore, we consider a marked image (Lena) and we compute the relative log-probability P_r for different values of M. It can be noticed in Figure 2(Left) that the curve obtained for the correct key has reached its maximum at 72 (the correct number of bits), and that it is almost constant (the values vary between -4 and 0) for a wrong key. We have the same behavior as for the second curve if we try to detect the mark in an unmarked image (Figure 2(Middle)).

[4] PSNR is the Peak Signal to Noise Ratio [6].

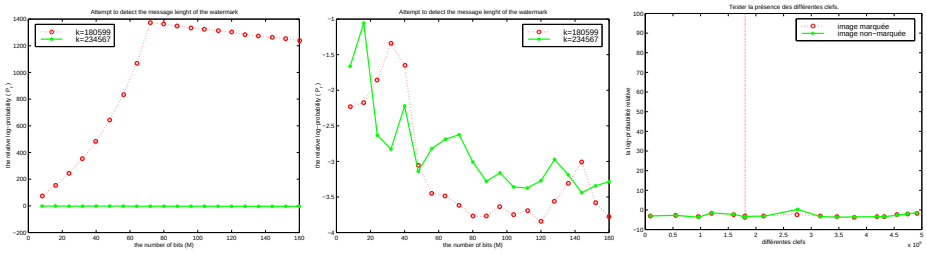

Fig. 2. An attempt to determine the correct number of bits in the watermark given only the key. Left: marked image. Middle: unmarked image. Right: An attempt to detect if a message was embedded with a given key.

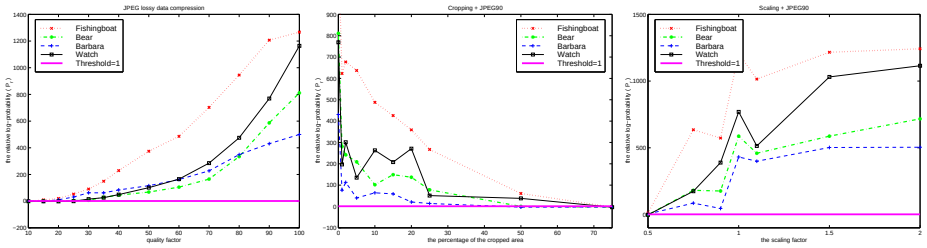

Fig. 3. The relative log-probabilities for some of the tests with Stirmark 3.1. Left: JPEG, Middle: Cropping + JPEG90 Right: Scaling + JPEG90.

6.2 Detecting the presence of the mark. The presented Bayesian approach can act concurrently to the message decoding, by estimating the probability that a watermark was generated given only the key. It can therefore be used either to extract the binary messages or to bind the watermark to the key.

Figure 2(Right) shows several attempts on the marked and unmarked images to detect and decode the message using different keys. We can see that the relative log-probabilities were below the threshold (generally between 0 and -4), excepting, when we used the correct key and the marked image.

6.3 Stirmark attacks. To test the robustness to different attacks of our algorithm, we applied the program Stirmark 3.1[5] of Fabien Petitcolas [6] on the marked images. The results are shown in Table 1. Figure 3 shows the behavior of the relative log-probability (P_r) in the case of some of the tests. The results show that the algorithm resist generally to the different attacks (excepting the random geometric distortion) Moreover, they show clearly that the Bayesian approach is more robust (the mark is detected for $P_r > 1$) than the message decoding ($ber = 0$) even if we accept[6] 1/72-3/72 bit error ratio ($ber \leq 5\%$).

[5] http://www.cl.cam.ac.uk/ fapp2/watermarking/stirmark/.

[6] These few bits can be recovered if we use error correcting codes.

Tests	$ber = 0$	$ber \leq 5\%$	$ber > 5\%$	$P_r \geq 1$	$P_r < 1$
JPEG	28/72	15/72	29/72	63/72	9/72
Enhancement± JPEG	60/66	3/66	3/66	63/66	3/66
Scaling ± JPEG	57/72	5/72	10/72	67/72	5/72
ARC ± JPEG	96/96	0/96	0/96	96/96	0/96
RAC ± JPEG	80/192	40/192	72/192	176/192	16/192
RACS ± JPEG	85/192	33/192	74/192	178/192	14/192
Cropping ± JPEG	57/108	21/108	30/108	92/108	16/108
LT ± JPEG	6/36	16/36	14/36	36/36	0/36
Shearing ± JPEG	18/72	20/72	34/72	71/72	1/72
RCR ± JPEG	33/60	12/60	15/60	60/60	0/60
Flip	6/6	0/6	0/6	6/6	0/6
RGD	0/6	0/6	6/6	0/6	6/6
CR	1/4	3/4	0/4	4/4	0/4

Table 1. The results of the tests with Stirmark 3.1. We used the following notations: *ber* is the *bit error ratio*; $ber \leq 5\%$ means in our case 1, 2 or 3 bit errors; ± JPEG means that the results on the corresponding row contains the tests with the modified image without compression and with compression JPEG of a quality factor 90. The enhancement techniques contains Gaussian and median filtering, frequency mode Laplacian removal and sharpening. LT are general linear transforms, RGD is a random geometric distortion. CR is a GIF format based color reduction (no color reduction for the two gray scale images). The variation of the parameters of different attacks is as follows: quality factor for JPEG between 10 and 90, scaling between 0.5 and 2; aspect ratio changes (ARC) between 0.8 and 1.2; angles for the rotation with auto-crop to remove zero padded regions (RAC) between -2 and 90; angles for the rotation with auto-crop to remove zero padded regions and rescale to the original size (RACS) between -2 and 90; the percentage of the cropped region between 1% et 75%; the percentage of the shearing along the x and y axes between 1% à 5% the number of the row and/or column regularly removed (RCR) between 1 and 17.

6.4 Different size of images. In our method we consider the DFT of the whole image and embed the mark inside a fixed frequency band. These frequencies were chosen as a compromise between visibility and robustness of the mark. As we work with absolute frequencies, the choice of the correct mid-frequency band depends on the image size. A frequency band adequate for a given size can be too low for large images, too high for small images. Therefore, for different sizes of images we have to consider different frequency bands. One possibility is to use normalized frequencies instead of absolute frequencies. As we do not know the original image size during the extraction, working with normalized frequencies requires fixed sized block at the embedding, which means that we have to divide the image into blocks or if the block is bigger than the image to pad it with zeros. Another possibility is to choose several (3-5 as a function of the image sizes we want to mark) fixed absolute frequency bands, and in function of the image size embed the mark in the more appropiate corresponding frequency band. During

Image	Lena	Bear	Benz
Inverting the found transformation	$P_r = -3.26$ $ber = 52.78\%$	$P_r = -3.07$ $ber = 45.83\%$	$P_r = -3.26$ $ber = 40.28\%$
Without inverting the found transformation	$P_r = 312.7$ $ber = 0\%$	$P_r = 68.6$ $ber = 1.39\%$	$P_r = 55.8$ $ber = 0\%$

Table 2. Some examples where the template does not resist to the JPEG compression (quality factor 50) and the found transform is not the correct one, however the mark was detected when we try to extract directly the mark.

the extraction, as the image can be cropped or scaled, therefore we do not know the original image size. We thus check each frequency band and estimate the relative log-probability P_r for each of them. The correct band will be the one for which $P_r > t$.

6.5 Using or not the recovered transformation. The template points require a stronger modification of the DFT magnitudes than the watermark since we want to insure that there will be local peaks. In order not to introduce visible artifacts, we choose higher frequencies for the template peaks than for the watermark. Several attacks such as JPEG compression with a high compression rate (low quality factor) affect first the high frequencies. Moreover, due to the properties of the spread spectrum, the watermark presents good robustness even if it is corrupted or partially considered. Therefore, it can happen that the template search algorithm is less robust than the watermark detection. In this case, if we invert the transformation (which is not necessarily correct or accurate) before detecting and decoding we may obtain a wrong message (see Table 2). However, if the attack contains no geometric transformation, the position of the watermark has not changed. Therefore, it is interesting to extract the mark first without looking for the geometric transformation and compute the relative log-probability. We then search for a geometric transform, invert it, extract again the mark and compare the new relative log-probability with the first one. We keep the message for which the relative log-probability was larger.

On the other hand, it is also important to estimate the geometric transformation even if the mark was detected without it. Indeed, if the transformation is not very large (small rotation for example), the mark can be detected without inverting the transformation, but with less accuracy (see Table 3).

6.6 Resistance of the Video watermarking. In order to test the robustness of the 3D DFT watermarking we use the video sequences compressed with MPEG-2. Figure 4 shows three marked frames for each of them. Both sequence have been encoded from PAL television images with a frame-rate of 25 frames/sec. The frames are of size 352×288 pixels, which is close to the 4/3 TV standard format ($352/288 = 3.667/3$). The first sequence, was 1450 frames long (58 s), and the second one, was 1749 frames long (1 min 10 s). Both video

Transformation	RC (-0.25)	Shearing	RCR (17,5)
Inverting the found transformation	$P_r = 524$ $ber = 0\%$	$P_r = 94.7$ $ber = 0\%$	$P_r = 158$ $ber = 0\%$
Without inverting the found transformation	$P_r = 1.58$ $ber = 16.67\%$	$P_r = 0.6$ $ber = 9.72\%$	$P_r = 69.09$ $ber = 1.39\%$

Table 3. Some examples where the mark was detected without the template, however the with template we have more accurate results.

Fig. 4. Three marked frames of both considered video sequences.

sequences were decompressed before performing any watermarking operation. The tests have been made using blocks of 16 frames long embedding the same watermark and the same template into each block. The results of the tests are shown in Table 4. Note that here each block was considered independently to test the presented algorithm on a maximum number of blocks. However, in the case of a video, generally the aspect ratio change and/or frame-rate change is the same along the whole video sequence. Therefore, it is sufficient to search for the template at the beginning of the video and once the template is detected, that is once the relative log-probability is positive, the obtained parameters can be used to invert the transformation for the next blocks.

6.7 Outliers Rejection. As the video is divided into consecutive chunks of fixed length and the same watermark is embedded into the 3D Discrete Fourier Transform magnitude of each block, for a given video one can compute an averaged $\boldsymbol{w}'$ or an "averaged" sequence of bits $\boldsymbol{b}$ in order to extract the message. However, some blocks, frames or video blocks can be more affected than others and therefore can be considered as "outliers". Using the Bayesian approach these "outliers" can be rejected easily before "averaging" and in this way the watermark extraction and message decoding becomes more robust (see Table 5).

Tests	$ber = 0$	$ber \leq 5\%$	$ber > 5\%$	$P_r \geq 1$	$P_r < 1$
MPEG-2	170/201	15/201	16/201	201/201	0/201
ARC+ MPEG-2	87/201	33/201	81/201	167/201	34/201
FRC+ MPEG-2	66/251	51/251	134/251	187/251	64/251
ARC+ FRC + MPEG-2	43/251	24/251	184/251	138/251	113/251

Table 4. The results for the 3D video watermarking approach. The aspect ratio changes (ARC) is 4/3 vs. 16/9; the frame-rate changes (FRC) is 25 vs. 30 fps. In the case of aspect ratio change of the frames, each frame is cropped and rescaled to the original size of the frame.

ARC+ FRC + MPEG-2	first video	second video
Averaging the results of all blocks	$ber = 2.88\%$	$ber = 0\%$
Averaging after outliers rejection	$ber = 0\%$	$ber = 0\%$

Table 5. Example (first video) where due to the outliers rejection the mark was more accurately detected than with simple averaging.

7 Conclusion

We have presented here a new approach for assessing the presence of a digital watermark in images and videos, by means of a Bayesian formulation that allows to compute the probability that a watermark was generated using a given key. The watermarking method itself relies on the discrete Fourier transform (DFT) of the image or of three dimensional chunks of video scene. The message is encoded in the spread spectrum signal using an owner key to ensure the security. The properties of m-sequences or Gold Codes were studied and exploited in order to generate this spread spectrum sequence.

In order to detect and invert the effect of an affine geometric transformation (rotations, scaling and/or aspect ratio change) or in the case of the video frame-rate changes, aspect-ratio modification and rescaling of frames, a point matching based template search algorithm was proposed.

Experimental results show on one hand the efficiency and the robustness of our watermark detection algorithm and, on the other hand, the behavior and the usefulness of the added Bayesian measure.

Acknowledgments

We are grateful to Shelby Pereira and Svyatoslav Voloshynovskyy for a number of interesting discussions on the subject, and to Dr A. Herrigel and Digital Copyright Technologies Switzerland for the ongoing collaboration. This work is financed by the European Esprit Open Microprocessor Initiative (project JEDI-FIRE) and by the Swiss Priority Program on Information and Communication

Structures (project Krypict) and is part of the European Patent application EU978107084.

References

1. I. Cox, J. Killian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for images, audio and video. In *Proceedings of the IEEE Int. Conf. on Image Processing ICIP-96*, pages 243–246, Lausanne, Switzerland, 1996.
2. F. Deguillaume, G. Csurka, J. J. K. Ó Ruanaidh, and T. Pun. Robust 3d dft video watermarking. In *IS&T/SPIE Electronic Imaging'99, Session: Security and Watermarking of Multimedia Contents*, San Jose, CA, USA, January 1999.
3. E. H. Dinan and B. Jabbari. Spreading codes for direct sequence CDMA and wideband CDMA cellular network. *IEEE Communications Magazine*, June 1998.
4. F. Hartung and B. Girod. Watermarking of uncompressed and compressed video. *Signal Processing*, 66:283–301, 1998.
5. J. R. Hernández, F. Pérez-González, J. M. Rodríguez, and G. Nieto. Performance analysis of a 2-D-multipulse amplitude modulation scheme for data hiding and watermarking of still images. *IEEE Journal on Selected Areas in Communications*, 16(4):510–523, May 1998.
6. M. Kutter and F. A. P. Petitcolas. A fair benchmark for image watermarking systems. In *Electronic Imaging '99, Security and Watermarking of Multimedia Contents*, volume 3657, pages 219–239, San Jose, CA, USA, January 1999.
7. N. Nikolaidis and I. Pitas. Robust image watermarking in the spatial domain. *Signal Processing*, 66:385–403, 1998.
8. Joe J. K. Ó Ruanaidh and Thierry Pun. Rotation, scale and translation invariant spread spectrum digital image watermarking. *Signal Processing*, 66(3):303–317, May 1998.
9. S. Pereira, J. J. K. Ó Ruanaidh, F. Deguillaume, G. Csurka, and T. Pun. Template based recovery of Fourier-based watermarks using Log-polar and Log-log maps. In *Int. Conference on Multimedia Computing and Systems, Special Session on Multimedia Data Security and Watermarking*, Juin 1999.
10. S. Pereira, J. J. K. Ó Ruanaidh, and T. Pun. Secure robust digital image watermarking using the lapped orthogonal transform. In *IS&T/SPIE Electronic Imaging'99, Session: Security and Watermarking of Multimedia Contents*, San Jose, CA, USA, January 1999.
11. R. L. Pickholtz, D. L. Schilling, and L. B. Milstein. Theory of spread spectrum communications – A tutorial. *IEEE Transactions on Communications*, COM-30(5):855–884, May 1982.
12. J. J. K. Ó Ruanaidh and W. J. Fitzgerald. *Numerical Bayesian Methods Applied to Signal Processing*. Series on Statistics and Computing. Springer-Verlag, 1996.
13. M. D. Swanson, B. Zhu, and A. H. Tewfik. Multiresolution scene-based video watermarking using perceptual models. *IEEE Journal on Selected Areas in Communications*, 16(4):540–550, May 1998.
14. A. Z. Tirkel, G. A. Rankin, R. G. van Schyndel, W. J. Ho, N. R. A. Mee, and C. F. Osborne. Electronic watermark. In *Dicta-93*, pages 666–672, Macquarie University, Sydney, December 1993.