Enhancing watermark robustness through mixture of watermarked digital objects [*]

Josep Domingo-Ferrer and Francesc Sebé
Dept. of Computer Engineering and Mathematics
Universitat Rovira i Virgili
Av. Països Catalans 26
E-43007 Tarragona, Catalonia, Spain
e-mail {jdomingo,fsebe}@etse.urv.es

Abstract

After the failure of copy prevention methods, watermarking stays the main technical safeguard of electronic copyright. There are many properties that a watermarking scheme should offer such as imperceptibility and robustness. The robustness property measures the resistance of the watermark against some attacks, which attempt to remove it partially or completely. Nowadays, many watermarking schemes exist each of them robust against a certain list of attacks but vulnerable to many others. It is not always easy to obtain new schemes that resist more and more attacks. This paper proposes general mixture techniques to combine the properties of several watermarking methods so as to obtain watermarked objects robust against most of the attacks survived by the combined methods.

Keywords: *Watermarking, Watermark mixture, Copyright protection, Robustness, Mixture functions.*

1 Introduction

The failure of sophisticated copy prevention systems like DVD [2] leaves copy detection, and more specifically watermarking, as the main solution for protecting the copyright of information in electronic format. Copyright protection involves owner identification and proof of ownership of the electronic information. Watermarking is also used for a number of applications beyond copyright protection (see [1]): broadcast monitoring, authentication, copy control, etc. In watermarking, the merchant embeds a mark in the copy sold and can later recover the mark from a redistributed copy to prove ownership, identify the redistributing buyer or do what the watermarking application requires.

A watermarking scheme is said to be *robust* if it survives common signal processing and geometric distortion attacks. Robustness should not be confused with tamper resistance: attacks to tamper resistance are of hostile nature and attempt to defeat the watermarking scheme using all information available on it; attacks to robustness involve general signal processing operations that the electronic information might accidentially suffer during its life cycle. For the specific case of images, the StirMark 3.1 [6] benchmark implements a number of robustness attacks against image marking systems such as JPEG compression, filtering, scaling, cropping and combinations of them. Current image watermarking proposals survive a limited number of these attacks.

1.1 Our contribution

Coming up with a watermarking method surviving all conceivable attacks may indeed be a difficult task. We explore in this paper ways to obtain increased robustness by mixing the outputs of several watermarking methods. Cocktail watermarking [4] can be regarded as a very specific attempt in this direction, since it combines positive and negative modulation; it is assumed that an attack damaging negative modulation results in a strenghtened positive modulation and conversely. Cocktail watermarking splits the transformed coefficients into two halves: the same Gaussian sequence is embedded into each half with different modulations. The approach presented in this paper is considerably more general and allows any set of watermarking methods to be mixed.

We will first discuss prior mixture, whereby a digital object is watermarked with different methods and a mixture of the watermarked objects is released. Posterior mixture will then be presented, which consists of mixing several attacked versions of the same watermarked digital object. It

[*] This work was partly funded by the Spanish CICYT under grant TIC-2001-0633-C03-01 and by the European Commission under project IST-2001-32012.

will be shown that prior mixture may result in a combination of the robustness properties of the watermarking methods being used. It will also be shown that posterior mixture may allow recovery of the embedded watermark, even if this watermark can no longer be recovered from each individual attacked version of the watermarked object. Note that prior or posterior mixtures are non-exclusive.

Section 2 describes prior mixture and illustrates the technique with an example. Section 3 does the same with posterior mixture. A conclusion and a few topics for future research are sketched in Section 4.

2 Prior mixture

Prior mixture is a general technique that allows a watermarked object to be obtained that combines the robustness properties of several watermarking schemes. No knowledge on the specific embedding and recovery algorithms is needed as they are used as a black box.

2.1 Description of the technique

2.1.1 Mark embedding and prior mixture

Let $E_1, \cdots, E_n$ be n different watermark embedding algorithms which can be used to embed a watermark M into the original digital object X. It is assumed in what follows that M contains some kind of redundancy (checksum, cyclic redundancy check, etc.), that allows to check its correctness or integrity. We then proceed as follows:

Algorithm 1 (Prior mixed embedding)

1. *The watermark M is embedded into X using algorithms $E_1, \cdots, E_n$ to obtain $X'_1, \ldots, X'_n$, where X'_i is the output of E_i.*
2. *A weight α_i is selected for each object X'_i, such that $0 \leq \alpha_i \leq 1$ and $\sum \alpha_i = 1$.*
3. *The watermarked mixed object is computed as*

$$X'_{premix} = f(\alpha_1, \cdots, \alpha_n, X'_1, \cdots, X'_n) \quad (1)$$

where f is a mixture function (see below).

Any mixture function can be used in Algorithm 1. However, sensible choices are an additive mixture

$$f(\alpha_1, \cdots, \alpha_n, X'_1, \cdots, X'_n) = \alpha_1 X'_1 + \cdots + \alpha_n X'_n \quad (2)$$

or a multiplicative mixture

$$f(\alpha_1, \cdots, \alpha_n, X'_1, \cdots, X'_n) = X'^{\alpha_1}_1 X'^{\alpha_2}_2 \cdots X'^{\alpha_n}_n \quad (3)$$

The above mixtures are componentwise between the semantically corresponding components of objects: for example, if the object is an image, components are pixels and the mixture amounts to averaging the color levels of corresponding pixels.

2.1.2 Mark recovery from a mixed object

Denote by $R_1, \cdots, R_n$ the watermark recovery algorithms corresponding to embedding algorithms $E_1, \cdots, E_n$ respectively. Let $\hat{X}$ be the object we want to recover the watermark from; if there have been attacks, $\hat{X}$ will not exactly match any watermarked object X'. The recovery procedure is as follows:

Algorithm 2 (Recovery from a mixed object)

1. *Run algorithms $R_1, \cdots, R_n$ on $\hat{X}$ and record the output of those algorithms, if any. Depending on the attacks suffered by $\hat{X}$ some algorithms may give no output.*
2. *Look for a correct watermark among the outputs of the recovery algorithms (the redundancy included in marks is checked for correctness). If all correct watermarks found have the same value, then recovery is successful. If there is no correct watermark or if there are several correct watermarks with different values, recovery fails.*

Note that mixing watermarked objects entails some amount of noise for each invidual watermarking method (E_i, R_i). In other words, when running recovery algorithm R_i, the effect of embedding algorithms E_j for $j \neq i$ is perceived as noise. Therefore, for prior mixture to be practical noise-robust watermarking methods must be used.

2.2 Example

In this example, prior mixture is demonstrated to combine the crop-proof and the scale-proof schemes for image watermarking presented in [8] and briefly recalled below. The resulting mixture stands both cropping and scaling attacks.

2.2.1 Crop-proof watermarking

This is a non-oblivious spatial-domain watermarking scheme. It uses the JPEG algorithm to decide the location and magnitude of the marks and relies on increasing/decreasing the color level of pixels to embed a watermark which consists of an encrypted and error-correcting coded binary sequence. The embedding algorithm has three input parameters: q is a JPEG quality level and is used as a robustness and capacity parameter; p is a Peak-Signal-to-Noise Ratio (PSNR) level and is used as an imperceptibility parameter; finally, k is a key for a cryptographic pseudo-random number generator which encrypts the sequence before embedding.

Under this algorithm, the mark is replicated and embedded into many pixels distributed all over the image and survives the following Stirmark 3.1 attacks: color quantization, low-pass filtering, JPEG compression and cropping. Other attacks are also survived, as long as their intensity is low (small scalings, small rotations, etc.).

2.2.2 Scale-proof watermarking

This is also a non-oblivious spatial-domain watermarking scheme. It divides the image into small tiles and embeds a bit of the watermark sequence inside each. The sequence is also encrypted and error-correcting coded prior to embedding. The embedding algorithm has three input parameters: p determines the tile size; r controls the inter-tile separation; finally, k is used to encrypt the sequence before embedding. Mark bits are also embedded by increasing/decreasing the color level of pixels.

This scheme survives color quantization, low-pass filtering, JPEG compression and scaling. A number of other attacks are also survived provided their intensity is low (small rotations, small croppings, etc.).

2.2.3 Mixed watermark robustness

The benchmark image Lena [7] was watermarked using the two aforementioned schemes, so that two watermarked versions were obtained. In both cases, the embedded watermark was the same 45-bit binary sequence. Prior image mixture was applied to mix the two watermarked versions of Lena. Additive and multiplicative mixtures with weights $\alpha_1 = \alpha_2 = 0.5$ were tried; in what follows, we report results only for additive mixture, which turned out to outperform multiplicative mixture for this particular example. Additive mixture with the above weights is actually the arithmetic average of color levels of pairs of pixels in the same position within images to be mixed.

The error correcting code (ECC) used in this experiment was a $(31, 5)$ dual Hamming binary code (with correcting capacity 7 errors). When attempting mark recovery from an attacked watermarked image, the average number of corrected errors at the decoding stage gives an indication of the vulnerability of the scheme against the attack. If the number of errors that must be corrected to reconstruct the watermark is low, then the scheme easily survives the attack; the higher the number of corrected errors after an attack, the more vulnerable is the scheme against the attack.

Table 1. Average no. of corrected errors at mark recovery for Lena (crop-proof method)

attack	crop-proof	mix crop-proof
JPEG 30	2.1	not survived
gaussian	0	2.3
sharpening	0	0
FMLR	1.9	not survived
median 3x3	0	1
cropping	0	0

Table 2. Average no. of corrected errors at mark recovery for Lena (scale-proof method)

attack	scale-proof	mix scale-proof
JPEG 30	0	2.8
gaussian	0	1
sharpening	0	3.2
FMLR	5.5	not survived
median 3x3	1.7	not survived
scaling	0	1.5

The following tables show the average number of errors corrected when recovering the watermark from the image Lena. Table 1 shows the average number errors corrected by the crop-proof recovery algorithm: the second column accounts for recovery from the crop-proof watermarked Lena (before mixing), while the third column refers to recovery from the mixed crop-proof and scale-proof Lena. Table 2 corresponds to errors corrected by the scale-proof recovery algorithm: its second column displays the average number of errors corrected when recovering a mark from the scale-proof watermarked Lena (before mixing); the third column refers to corrected errors in the recovery from the mixed crop-proof and scale-proof Lena.

It can be seen from Tables 1 and 2 that the result of mixing both schemes is a non-oblivious image watermarking scheme robust against color quantization, low-pass filtering, JPEG compression, cropping and scaling. Thus we have succeeded in combining resistance against cropping attacks with resistance against scaling attacks. Of course, the amount of noise tolerated by the mixture of both schemes is lower than the amount that would be tolerated by each scheme individually and some filters like FMLR are no longer survived.

The experiment above was repeated with other benchmark images in [7], and the results were similar to those obtained with Lena.

2.2.4 Mixed watermark imperceptibility

Imperceptibility is a very important feature of a watermarking scheme. It refers to the extent to which the image quality

Table 3. PSNR of watermarked vs original images

	crop-proof	scale-proof	mixed
Lena	38	41.12	40.59
Baboon	36.7	36.52	36.76

is preserved after the mark has been embedded. The Peak Signal-to-Noise Ratio (PSNR) between the original and the watermarked images is one common way to measure imperceptibility.

Table 3 shows how, after mixture, image quality does not decrease but stays similar or even higher than quality of watermarked images input to mixture. Table rows correspond to images Lena and Baboon [7]. Table columns correspond to the three watermarking possibilities: crop-proof only, scale-proof only or additive mixture of both methods. For each image, the PSNR of the three watermarked versions vs the original image is given. It is noteworthy that the PSNR of the mixed image can even be higher than the PSNR of images watermarked with a single method.

3 Posterior mixture

Posterior mixture is a technique usable if the following assumptions hold:

A1. Several attacked versions $\hat{X}_1, \cdots, \hat{X}_m$ originating from the same watermarked digital object X' are available, where the watermarking method used and the embedded watermark are the same for all attacked versions. The difference between versions is only caused by the attacks they have undergone.

A2. None of $\hat{X}_1, \cdots, \hat{X}_m$ separately allows recovery of the common embedded watermark.

A3. It must be possible to find a one-to-one mapping between semantically corresponding components of $\hat{X}_i$ and $\hat{X}_{i+1}$, for $i = 1$ to $m - 1$. Note that some attacks may render fulfilling this assumption difficult or even infeasible. For example, let objects be images; then components are pixels and mapping semantically equivalent pixels may require undoing scaling attacks, rotation attacks, mapping cropped images with the corresponding parts of uncropped images, etc.

The procedure is as follows:

Algorithm 3 (Posterior mixed recovery)

1. *Mix the attacked watermarked objects, by computing*

$$\hat{X}_{postmix} = f(\beta_1, \cdots, \beta_m, \hat{X}_1, \cdots, \hat{X}_m)$$

where f is a componentwise mixture function (mixing semantically corresponding components, see Assumption A3 above) and β_j, for $j = 1, \cdots, m$ are weights such that $0 \leq \beta_j \leq 1$ and $\sum \beta_j = 1$.

2. *Use the recovery algorithm of the common watermarking method to recover the embedded watermark from $\hat{X}_{postmix}$.*

Algorithm 3 must be regarded as a last chance to repair an otherwise unrecoverable attacked watermark. Posterior mixture can be used as a second line of defense in combination with prior mixture, *i.e.* prior mixture can be used before the attacks happen and posterior mixture after the attacks have happened: in this case, the attacked $\hat{X}_1, \cdots, \hat{X}_m$ would originate from the same prior mixed object X'_{premix} (see Expression (1)).

3.1 Example

This example will illustrate posterior mixture to recover a watermark from several attacked versions of an image watermarked using the oblivious scheme described in [9]. Oblivious watermarking does not require the original image for mark recovery, but tends to be somewhat less robust than non-oblivious watermarking. Posterior mixture can be a valuable option to supplement robustness of oblivious methods.

3.1.1 An oblivious watermarking method

The oblivious scheme presented in [9] embeds a watermark consisting of a binary sequence that has previously been coded using an error-correcting code. Using a secret key k as seed, the embedding algorithm pseudo-randomly places one non-overlapped square tile over the image for each bit of the watermark. Then, the corresponding bit is embedded into each pixel of the tile. The way to embed a bit into a pixel is as follows. First of all, the gray-scale range is divided into subintervals and then these subintervals are consecutively labeled as “0” or “1”. Then, to embed the bit inside a pixel, its gray-scale level is modified so that it falls into a subinterval labeled with the corresponding digital value.

This scheme survives Stirmark 3.1 attacks like low pass filtering, JPEG compression (down to quality 30), scaling, and other geometric distortion attacks.

3.1.2 Robustness results

A sequence of 35 bits was embedded into the benchmark image Lena. The embedded sequence was a codeword of a $(31, 5)$ dual Hamming binary code.

Two attacks were performed on the watermarked image: the first one consisted of JPEG compression with quality 20, and the second one was a sharpening filter. From none of both attacked images could the watermark be recovered.

Posterior additive mixture with weights $\beta_1 = \beta_2 = 0.5$ was used to mix both attacked images. In other words, the arithmetic average of color levels for semantically corresponding pixels in the attacked images was computed; since neither compression nor sharpening attacks alter the size nor the orientation of images, semantically corresponding pixels are those occupying the same position in both images. The watermark was recoverable from the posterior mixed image, with an average number of 2.5 corrected errors per codeword, well below the correcting capacity of the $(31, 5)$ dual Hamming binary code (7 errors).

Exactly the same experiment was successfully repeated with other benchmark images, like Skyline_arch and Bear [7]. For those images, the average number of corrected errors per codeword were, respectively, 6 and 5.7, which are already closer to the correcting capacity of the code. Thus, the effectiveness of posterior mixture depends on the particular image, method and attacks being dealt with.

4 Conclusion and future research

Two general approaches have been presented which allow watermark robustness to be enhanced. Rather than developing more robust watermarking algorithms from scratch, the idea is to build on the robustness of existing algorithms. One of the techniques proposed operates before attacks happen and the other after attacks have happened. Both techniques can be combined and be used with a broad range of watermarking methods. Future research will be directed at finding new mixture functions and weight assignments that maximize robustness against specific attacks.

References

[1] I. J. Cox, M. L. Miller and J. A. Bloom, "Watermarking applications and their properties", in *International Conference on Information Technology: Coding and Computing*. Los Alamitos CA: IEEE Computer Society, 2000, pp. 6-10.

[2] `http://www.lemuria.org/DeCSS`

[3] M. Kutter and F. A. P. Petitcolas, "A fair benchmark for image watermarking systems", in *Security and Watermarking of Multimedia Contents*, vol. 3657. San José CA: Soc. for Imaging Sci. and Tech. and Intl. Soc. for Optical Eng., 1999, pp. 226-239.

[4] C.-S. Lu, H.-Y. Mark Liao, S.-K. Huang and C.-J. Sze, "Highly robust image watermarking using complementary modulations", in *Information Security-ISW'99*, LNCS 1729. Berlin: Springer-Verlag, 1999, pp. 136-153.

[5] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, "Attacks on copyright marking systems", in *2nd International Workshop on Information Hiding*, LNCS 1525. Berlin: Springer-Verlag, 1998, pp. 219-239.

[6] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, *StirMark v3.1*
`http://www.cl.cam.ac.uk/`
`~fapp2/watermarking/stirmark/`

[7] `http://www.cl.cam.ac.uk/~fapp2/`
`watermarking/benchmark/image_database.html`

[8] F. Sebé, J. Domingo-Ferrer and J. Herrera. "Spatial-domain image watermarking robust against compression, filtering, cropping and scaling". In *Information Security Workshop'2000*, LNCS 1975. Berlin: Springer-Verlag, 2000, pp. 44-53.

[9] Francesc Sebé and Josep Domingo-Ferrer. "Oblivious image watermarking robust against scaling and geometric distortions", In *Information Security Conference'2001*, LNCS 2200. Berlin: Springer-Verlag, 2001, pp. 420-432.