Oblivious Image Watermarking Robust against Scaling and Geometric Distortions*

Francesc Sebé and Josep Domingo-Ferrer

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

Abstract. Watermarking stays the main technical safeguard of electronic copyright. This paper presents the first public-domain oblivious watermarking scheme for images which survives scaling and geometric distortion attacks. Previous proposals are either proprietary, non-oblivious or require scaling or geometric distortion to be undone prior to mark recovery, which may not be practical in oblivious watermarking. The new scheme uses a tile-based embedding technique that allows mark recovery from a scaled or geometrically distorted watermarked image. Other properties of the scheme presented here are that it operates on the spatial domain, supports multiple marking and does not require previous knowledge of the embedded copyright sequence. The latter property makes the proposed watermarking suitable for fingerprinting purposes.

Keywords: Oblivious watermarking, Multimedia copyright protection.

1 Introduction

The failure of sophisticated copy prevention systems like DVD [12] leaves copy detection, and more specifically watermarking, as the main solution for protecting the copyright of images or information in electronic format. In watermarking, the merchant embeds a mark in the copy sold and can later recover the mark from a redistributed copy to prove ownership or identify the redistributing buyer. Published watermarking schemes can be divided into oblivious and non-oblivious; the first class is more convenient for large-scale mass protection of digital content because it does not require the original content for mark recovery, while the second class does.

Oblivious watermarking offers a greater organizational flexibility and is better adapted to distributed copy detection than non-oblivious watermarking. For example, it enables the merchant to delegate copy detection to a set of agents distributed over the Internet, who can recover marks from intercepted redistributed content without having been entrusted with the original content. Such an arrangement minimizes disclosure of the original unprotected content (which

* This work is partly funded by the Spanish CICYT under contract no. TEL98-0699-C02-02.

G.I. Davida and Y. Frankel (Eds.): ISC 2001, LNCS 2200, pp. 420–432, 2001.
© Springer-Verlag Berlin Heidelberg 2001

stays only known to the merchant) and also minimizes storage requirements (agents do not have to store the original version of all digital content they can come across of).

Commercial oblivious watermarking schemes surviving a broad range of manipulations (*e.g.* Digimarc [4]) tend to be based on propietary algorithms not available in the literature.

1.1 Plan of This Paper

Section 2 is an overview of the literature on oblivious watermarking. Section 3 highlights the most outstanding features of our contribution in comparison to proposals in the literature. In Section 4, our procedure for mark embedding is described. Section 5 deals with the procedure for mark recovery. Parameter choice is discussed in Section 6. Empirical results on imperceptibility are given in Section 7. Empirical results on robustness are given in Section 8. Conclusions and topics for future research are summarized in Section 9.

2 Background on Oblivious Watermarking

There are two shortcomings affecting oblivious watermarking systems in the literature:

- Many published proposals require the embedded sequence to be given as an input to the mark detection procedure.
- No proposal in the literature embeds marks so that they can survive scaling and/or geometric distortion attacks.

In the two subsections below, we explain the two shortcomings above in more detail.

2.1 Systems That Require Knowledge of the Embedded Sequence

Many recently published oblivious schemes require previous knowledge of the embedded sequence before mark detection. This requirement makes mark recovery more robust but less flexible as the merchant needs to know beforehand which sequence she is looking for. Assuming such knowledge is definitely unrealistic if watermarking is used for fingerprinting (where the merchant embeds a different serial number or buyer ID in each copy being sold [11,5]).

Examples of schemes with this problem are [10,2,7,8]. In these proposals, the watermark takes the form of a Gaussian or binary pseudo random sequence s which is embedded in some transform domain. Let $C = \mathcal{T}(I)$, where $\mathcal{T}$ denotes some transform and C are the transform coefficients of the original unmodified image I. A subset c of C is modified to embed the watermark. Let $C = c \cup \overline{c}$, where $c \cap \overline{c} = \emptyset$. Denoting by $\mathcal{E}$ the watermark embedding function, the overall embedding operation can be expressed as

$$C = \mathcal{T}(I) \qquad \hat{c} = \mathcal{E}(c, s) \qquad \hat{C} = \hat{c} \cup \overline{c} \qquad \hat{I} = \mathcal{T}^{-1}(\hat{C})$$

Let $\tilde{I} = \hat{I} + N$ be the image in which the presence of the watermark is tested, where N is some noise that can appear between mark embedding and mark detection. The detection operation can be expressed as

$$\tilde{C} = \mathcal{T}(\tilde{I}) \qquad \tilde{s} = \mathcal{D}(\tilde{c}) \qquad s_d = \frac{s^T \tilde{s}}{\mid s \mid\mid \tilde{s} \mid}$$

where $\mathcal{D}$ is a detector function and s_d is the detection statistic ($-1 \leq s_d \leq 1$) which is a measure of the normalized correlation of the embedded and detected signature sequences. In these schemes, an image $\tilde{I}$ is considered to contain the watermark s if s_d is greater than a fixed threshold. Of course, computing s_d requires previous knowledge of s.

2.2 Scaling and Geometric Distortion Attacks

To our best knowledge, robustness against scaling and geometric distortion attacks is not achieved by any published oblivious scheme. Some previous schemes assume that such attacks can be undone prior to mark recovery [10,7]. Undoing them requires knowledge of the original image which turns those schemes into non-oblivious ones.

In [1] an iterative search technique is used to cope with geometric attacks. The search technique seeks to emulate the inverse operation of the attack. It consists of running the mark recovery algorithm after trying various inverse operations until the bit error rate of the hidden bits drops dramatically or a high correlation with the original watermark is obtained. Thus, even if this system does not generally require knowledge of the embedded mark for recovery, it definitely requires such knowledge in order to survive geometric attacks. Furthermore, the search technique used to undo such attacks is too expensive and cannot be applied to random distortion attacks where the inverse operation is unknown.

In [3], image pixels are permuted using a pseudo random seeded permutation. Then the permuted image is divided into blocks and information is embedded by modifying the standard deviation of gray (or color) levels of the pixels in each block. Finally, the permutation is undone. During mark recovery, the image is again permuted and the embedded watermark is extracted. The problem is that, if the watermarked image has undergone geometric distortion, permuting it and dividing it into blocks will yield a block partition completely different from the one used in the embedding process, so that the recovered mark will be incorrect.

In other oblivious schemes, geometric attacks are left for future research [2, 8] or are not even mentioned [6,3]. It is actually unclear whether the latter two proposals can be termed robust, because they are very sensitive to geometric distortion.

We next sketch the fundamentals of the widely used method [6], so that its vulnerability to geometric distortion attacks can be better understood. This method uses a spread spectrum technique to obtain an oblivious watermarking system. The embedding and recovery procedures are as follows:

Embedding. The copyright information to be embedded is a binary sequence $a_j, \quad a_j \in \{-1, 1\}$. This discrete signal is spread by a large factor cr, called chip-rate, to obtain the sequence $b_i = a_j, \quad j \cdot cr \leq i < (j+1) \cdot cr$. The spread sequence b_i is amplified by a locally adjustable amplitude factor $\alpha_i \geq 0$ and is then modulated by a binary pseudo-noise sequence $p_i, \quad p_i \in \{-1, 1\}$. If v_i is the original signal to be protected, the resulting watermarked signal is $\tilde{v}_i = v_i + \alpha_i \cdot b_i \cdot p_i$.

Recovery. Mark recovery is performed by demodulating the watermarked signal with the same pseudo-noise signal p_i that was used for embedding, followed by summation over the window for each embedded bit, which yields the correlation sum s_j for the j'th information bit $s_j = \sum_{i=j \cdot cr}^{(j+1) \cdot cr - 1} p_i \cdot \tilde{v}_i$. The sign of s_j is interpreted as the embedded bit a_j. To generate a correct result, each pixel $\tilde{v}_i$ must be demodulated by the corresponding p_i of the random sequence. This requirement turns this scheme very vulnerable to geometric distortion attacks which can cause the syncronization between sequences $\tilde{v}_i$ and p_i to be lost.

3 Our Contribution

In addition to being more robust, the oblivious watermarking scheme presented in this paper is simpler than the aforementioned ones in that it operates in the spatial domain rather than in transformed domains (DCT, wavelet). It does not require prior knowledge of the embedded sequence and survives all kinds of scaling and moderate geometric distortion attacks (*e.g.* row and column removal, shearing, random distortion and small cropping and rotation) without requiring the original image for mark recovery. It is also robust against most filtering and compresion attacks performed by the StirMark 3.1 benchmark (see [9] and below).

4 Mark Embedding

Without loss of generality, we will assume a monochrome image in what follows; for RGB color images, watermarking is independently done for each color plane. Let the original image be $X = \{x_i : 1 \leq i \leq n\}$, where x_i is the color level of the i-th pixel and n is the number of pixels in the image. Let x_i take integer values between 0 and $MAXCOLOR$, so that the lower x_i, the darker is the color level. Let $dt \in [0, MAXCOLOR]$ be a threshold such that all color levels x_i below dt visually appear as dark.

An algorithm is next given whereby the merchant can compute pixel visual components, that is, the perceptual value of pixels. This value is an estimate of the maximum subperceptual increment/decrement that each pixel can accomodate. The idea underlying Algorithm 1 is that dark pixels and those pixels in non-homogeneous regions are the ones that can best accomodate embedded information while minimizing the perceptual impact. In Algorithm 1 below, let

lb_1 and ub_1 be integer values $lb_1 < ub_1$ that are used as parameters to bound the variation of pixel color values. For a given $MAXCOLOR$, suitable values for dt, lb_1 and ub_1 are empirically chosen; for example, for $MAXCOLOR = 255$ (8-bit color level), a good choice is $dt = 70$, $lb_1 = 2$, $ub_1 = 11$.

Algorithm 1 (Visual components(dt, lb_1, ub_1))

1. *For $i = 1$ to n do:*
   a) *Compute $m_i := \max_j |x_i - x_j|/2$, for all pixels j which are neighbors of pixel i on the image (there are up to eight neighbors); m_i can be regarded as a kind of discrete derivative at pixel i. To bound the value of m_i between lb_1 and ub_1, perform the following corrections:*
      i. *If $m_i > ub_1$ then $m_i := ub_1$.*
      ii. *If $m_i < lb_1$ then $m_i := lb_1$.*
   b) *Compute the darkness of the i-th pixel as $d_i := (dt - x_i) * ub_1/dt$ if $x_i < dt$ and $d_i := 0$ otherwise. A pixel is considered as dark if its color level is below dt. The value of d_i lies between 0 and ub_1.*
   c) *Compute the preliminary visual component of the i-th pixel as $v_i := \max(m_i, d_i)$.*
2. *For $i = 1$ to n compute the final visual component of the i-th pixel as $V_i := \max_j v_j$, for all pixels j which are neighbors of i on the image plus the pixel i itself.*

The higher V_i for a pixel, the less perceptible are changes in that pixel. The merchant is now ready to embed a copyright binary sequence in the image. The embedding process is governed by a key k known only to the merchant and is based on incrementing/decrementing the pixel color level. The absolute variation for a pixel is always less that its visual component. Integers lb_2 and ub_2 in the algorithm below are parameters such that $lb_2 < ub_2$. Given $MAXCOLOR$, a suitable value is determined for these parameters (see also Section 6). For example, $lb_2 = 10$ and $ub_2 = 13$ are good choices for $MAXCOLOR = 255$.

Algorithm 2 (Mark embedding(k, lb_2, ub_2))

1. *If ε is the binary sequence to be embedded, encode ε using an error-correcting code (ECC) to obtain the encoded mark E.*
2. *Using the key k as seed, pseudo-randomly place $|E|$ non-overlapped square tiles R_i over the image, where $|E|$ is the bitlength of E. Tile size is also determined by k.*
3. *Using k as seed, pseudo-randomly assign a value a_i between lb_2 and ub_2 to tile R_i, for $i = 1$ to $|E|$.*
4. *To embed the i-th bit e_i of the mark E in R_i:*
   a) *Divide the color level interval $[0, MAXCOLOR]$ into subintervals of size a_i.*
   b) *Label consecutive subintervals alternately as "0" or "1".*

c) For each pixel x_j in R_i:

i. If x_j lies in a subinterval labeled e_i, bring it as close as possible to the interval center by increasing or decreasing x_j no more than V_j.

ii. If x_j lies in a subinterval labeled $\bar{e_i}$, bring it as close as possible to the nearest neighbor interval center (neighbor intervals are labeled e_i) by increasing or decreasing x_j no more than V_j.

5 Mark Recovery

Upon detecting a redistributed image $\hat{X}$, $\hat{\varepsilon}$ can be recovered as follows, provided that the length $|E|$ of the embedded mark and the secret key k used for embedding are known (the merchant should know these parameters).

Algorithm 3 (Mark recovery(k, lb_2, ub_2))

1. *Using the key k as seed, pseudo-randomly place $|E|$ non-overlapped square tiles R_i over the image (again, tile size is also determined by k). Also using k as seed, pseudo-randomly assign a value a_i between lb_2 and ub_2 to tile R_i, for $i = 1$ to $|E|$. (Tiling done in this step is analogous to tiling done during mark embedding).*
2. *To recover the i-th bit $\hat{e}_i$ of $\hat{E}$ from R_i:*
   a) Divide the color level interval $[0, MAXCOLOR]$ into subintervals of size a_i.
   b) Label consecutive subintervals alternately as "0" or "1".
   c) Let $ones := 0$ and $zeroes := 0$.
   d) For each pixel x_j in R_i:
      i. If x_j lies in a subinterval labeled "1", then $ones := ones + 1$
      ii. If x_j lies in a subinterval labeled "0", then $zeroes := zeroes + 1$
   e) If $ones > zeroes$ then $\hat{e}_i := 1$; if $ones < zeroes$ then $\hat{e}_i := 0$; otherwise $\hat{e}_i$ is an erasure.
3. *Decode $\hat{E}$ with the same ECC used for embedding to obtain $\hat{\varepsilon}$.*

6 Parameter Choice

In Algorithm 1 (visual components), suitable values for parameters dt, lb_1 and ub_1 should be empirically chosen for a given $MAXCOLOR$. Suitability depends on visual perception and robustness considerations. We have suggested above a good choice for $MAXCOLOR = 255$, namely $dt = 70$, $lb_1 = 2$ and $ub_1 = 11$; for other values of $MAXCOLOR$, a rule of thumb of is to scale that choice by $MAXCOLOR/255$. We discuss below other parameters related to Algorithm 2 (embedding) and Algorithm 3 (recovery).

6.1 On the Size of Tiles

At Step 2 of the mark embedding algorithm, $|E|$ square tiles are randomly placed over the image. From the point of view of robustness, the tile size must be large enough so that each bit is embedded in a sufficient number of pixels. However, the requirement that all $|E|$ tiles should be placed non-overlappingly limits the maximum tile size, which decreases as $|E|$ increases.

An additional consideration is imperceptibility. Better imperceptibility is gained if neighboring tiles are separated by a band of unmodified pixels. This further limits the tile size.

6.2 On the Width of Color Level Subintervals

The size a_i in which we divide the color level interval is a tradeoff between robustness and imperceptibility:

- Making such intervals narrow means that, during the mark embedding algorithm, the variation applied to pixels to mark them is low, which leads to better imperceptibility. The drawback of narrow intervals is a loss of robustness, because even small noise can easily shift a color level to a neighboring subinterval, which can cause an incorrect bit to be recovered.
- Larger values a_i yield higher robustness, but moving the color level of a pixel to a neighboring subinterval can be perceptible.

Thus, given $MAXCOLOR$, the interval $[lb_2, ub_2]$ where a_i randomly takes values has its lower bound lb_2 limited by robustness and its upper bound ub_2 limited by imperceptibility.

7 Imperceptibility Assessment

The scheme was implemented with parameter values $MAXCOLOR = 255$, $dt = 70$, $lb_1 = 2$, $ub_1 = 11$, $lb_2 = 10$, $ub_2 = 13$. The error-correcting code used was a dual binary Hamming code $DH(31, 5)$. If there 7 or less errors in a codeword, this code guarantees correction; for more than 7 errors, correction is not guaranteed.

The following images from [13] were tried: Lena, Bear, Baboon and Peppers. A 30-bit long mark ε was used, which needed six codewords of the dual Hamming code and resulted in an encoded E with $|E| = 31 \times 6 = 186$ bits. For Lena, a version of size 512×512 pixels was used and the length of the tile side was randomly chosen between 11 and 31; for the other images, a similar proportion between image size and tile size was maintained. For all images, tiles were placed so that neighboring tiles were separated by a band of unmodified pixels at least one pixel wide.

Figure 1 shows the original and the marked Lena; the Peak Signal-to-Noise Ratio (PSNR) between both images is as high as 41.16 dB.

Fig. 1. Left, original Lena. Right, Lena after embedding a 30 bit length mark (PSNR=41.16 dB)

7.1 Multiple Marking

A useful feature of the presented algorithm is that multiple marking is supported. Up to three consecutive markings on the same image are possible without substantial perceptual degradation nor loss of robustness. For example, the content creator M_1 can mark an image and sell the marked image to a distributing company M_2 which re-marks the image with its own mark, re-sells it to a retailer M_3, who re-marks the image again before selling it to the end consumer. Each of M_1, M_2, M_3 can recover their embedded watermark by using the same key they used at embedding time.

Each of the four test images was marked three times and the results shown in Table 1 were obtained. The table shows the PSNR between each original image and the successive marked versions of it; the more markings, the lower is PSNR, which means more degradation of perceptual quality. The table also shows the average number of errors corrected per codeword of the encoded copyright sequence during mark recovery; this average is computed over the six codewords used to encode the mark. The number of errors per codeword increases with the number of consecutive markings and puts a strict limitation on how many times the image can be re-marked: if a codeword recovered from the marked image contains more errors than the correcting capacity of the ECC being used (7 errors in our implementation), then the mark might not be correctly recovered.

To illustrate the effects on imperceptibility, Figure 2 shows Lena after three consecutive markings.

8 Robustness Assessment

Some general considerations regarding robustness in front of the various kinds of attacks are as follows:

Table 1. Variation of perceptual quality and error rate for consecutive markings

	1st marking		2nd marking		3rd marking	
Image	PSNR (dB)	Errors	PSNR (dB)	Errors	PSNR (dB)	Errors
Lena	41.2	1	38.2	1.16	36.4	2.6
Peppers	39.4	0.16	36.2	1.83	34.5	2.6
Bear	39	0.16	35.4	1.33	33.7	3.1
Baboon	37.5	0	34.2	0.8	32.4	4

Fig. 2. Lena after three consecutive markings

- After an attack, a bit is correctly recovered if a majority of correct mark bits are still inside the corresponding tile.
- Scaling attacks are survived by placing tiles in positions relative to the image size. In this way, even if the image size varies, each tile still contains original mark bits.
- Unless tiles are very small, other attacks like row and column removal, shearing, cropping and rotation will only succeed if most pixels in the tile suffer a variation so large that it leads to visual degradation.

The base test of the StirMark 3.1 benchmark [9] was used to evaluate robustness on the marked versions of the four test images. The same parameter values listed at the beginning of Section 7 were taken and the following manipulations were survived:

1. Color quantization
2. Most low-pass filtering manipulations. More specifically:
   a) Gaussian filter (blur)
   b) Median filter (2×2, 3×3 and 4×4).
   c) Linear filter
3. JPEG compression for qualities 90 down to 30.

4. All StirMark scaling attacks (scale factors from 0.5 to 2).
5. All StirMark aspect ratio modification attacks.
6. All StirMark row and column removal attacks.
7. All StirMark shearing attacks.
8. Small rotations with and without scaling from -2 to 2 degrees.
9. Small cropping up to 2%.
10. StirMark random bend.

Robustness against different attacks has been measured taking into account the average number of errors corrected per codeword of the encoded copyright sequence. Table 2 shows the average number of errors per codeword when recovering the mark from the watermarked image Lena after an attack with low-pass filters and color quantization. The proposed scheme is very robust against this kind of attacks, since the average number of errors is well below the error-correcting capacity of the ECC used (7 errors).

Table 2. Average number of errors per codeword in mark recovery after low-pass filtering and color quantization (Lena)

Attack	Errors/codeword
Gaussian	3.16
Median (2×2)	1
Median (3×3)	1.5
Median (4×4)	1.83
Linear	2.83
Color quantization	2

Table 3 shows the average number of errors per codeword after compressing the watermarked image Lena with different JPEG quality levels. It can be seen that the average number of errors grows linearly with the compression rate (inverse of JPEG quality level). Figure 3 shows the watermarked Peppers image after a compression attack at JPEG quality 30; the mark is still recoverable.

Table 4 shows the average number of errors per codeword when recovering the mark from the watermarked image Lena after scaling attacks. It can be seen that very few errors are introduced when the scaling factor is greater than 0.90. The number of errors increases linearly as the scaling factor decreases toward 0.50. For Lena, recovery is still correct after 0.50 scaling in spite of there being an average 5.83 errors/codeword (see Figure 4). If there was some codeword with more than 7 errors, recovery could fail; this does not happen for any of the images tried, but could happen for other images after a 0.50 scaling attack.

Another family of StirMark attacks are aspect ratio modifications in the x and y directions with scaling factors between 0.8 and 1.2. The average number of errors introduced by such attacks is rather low (about 1.5 errors/codeword). Thus, the damage inflicted by these attacks is similar to the damage caused by moderate scaling attacks.

Table 3. Average number of errors per codeword in mark recovery after JPEG compression (Lena)

JPEG quality	Errors/codeword
90	1.16
80	1.83
70	2.16
60	2.66
50	3.66
40	4.83
30	5.16

Fig. 3. Marked Peppers compressed at JPEG quality 30 (mark still recoverable)

Table 4. Average number of errors per codeword in mark recovery after scaling (Lena)

Scaling factor	Errors/codeword
0.50	5.83
0.75	2.5
0.90	1.33
1.10	1.33
1.50	1.33
2.00	0.5

Table 5 shows robustness results after rotation with and without scaling on the watermarked Lena. In rotation without scaling, the rotated image is cropped so that it completely fills a horizontal frame (the resulting image is smaller than the original); in rotation with scaling, the rotated image is cropped and then magnified so that the horizontal frame being filled is of the same size as the original image. For small rotations, the average number of errors per codeword

Fig. 4. Left, Lena after a 50% scaling attack. Right, Lena after the StirMark random bend attack. The mark is correctly recovered from both images.

is similar with and without scaling; for each rotation angle, Table 5 gives the joint average of the average number of errors per codeword in both cases.

Table 5. Average number of errors per codeword in mark recovery after rotation with and without scaling (Lena)

Rotation (degrees)	Errors/codeword
0.25	1.65
0.50	1.87
0.75	2.54
1.00	2.91
2.00	3.45

Small croppings and the StirMark random bend are also survived for the four images tried. For Lena, 1% cropping is survived with an average of 1.66 errors per codeword and 2% cropping with 2.16 errors. Also for Lena, the StirMark random bend attack is survived with an average of 3.33 errors per codeword (see Figure 4).

9 Conclusions and Future Research

Oblivious watermarking is attractive because it allows the content owners to delegate redistributor tracing to agents that need not be entrusted with the original content. However, oblivious watermarking methods in the literature fail to survive geometric distortions. We have presented here the first oblivious system which offers robustness against a comprehensive range of such attacks. In

addition, the proposed system operates in the spatial domain and is thus conceptually simpler than most previous oblivious systems.

Future research will be devoted to further enhancement of the robustness of oblivious watermarking systems. A desirable achievement would be to add all StirMark croppings to the list of survived attacks.

Another topic that should be investigated both for oblivious and non-oblivious watermarking is tamper-proofness. Benchmarks like StirMark are useful to evaluate robustness against standard signal processing attacks, but they do not deal with tamper-proofness, *i.e.* with attacks that exploit knowledge of the algorithm internal operation. If watermarking is to conform to Kerchhoff's assumption of algorithms being public-domain, tamper-proofness becomes a relevant issue.

References

1. F. Alturki and R. Mersereau, "Robust oblivious digital watermarking using image transform phase modulation", in *International Conference on Image Processing - ICIP'2000*, IEEE Signal Processing Society, 2000.
2. M. Caramma, R. Lancini, F. Mapelli and S. Tubaro, "A blind & readable watermarking scheme for color images", in *International Conference on Image Processing - ICIP'2000*, IEEE Signal Processing Society, 2000.
3. P.-M. Chen, "A robust digital watermarking based on a statistical approach", in *International Conference on Information Technology: Coding and Computing - ITCC'2000*, IEEE Computer Society, 2000, pp. 116-121.
4. Digimarc, `http://www.digimarc.com`
5. J. Domingo-Ferrer and J. Herrera-Joancomartí, "Short collusion-secure fingerprints based on dual binary Hamming codes" *Electronics Letters*, 36(20): 1697-1699, Sep. 2000.
6. F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video", *Signal Processing*, 66(3): 283-301, May 1998.
7. C.-S. Lu and H.-Y. Mark Liao, "Oblivious cocktail watermarking by sparse code shrinkage: a regional- and global-based scheme", in *International Conference on Image Processing - ICIP'2000*, IEEE Signal Processing Society, 2000.
8. C.-S. Lu and H.-Y. Mark Liao, "An oblivious and robust watermarking scheme using communications-with-side-information mechanism", in *International Conference on Information Technology: Coding and Computing - ITCC'2001*, IEEE Computer Society, 2001, pp. 103-107.
9. F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, *StirMark v3.1* `http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark/`
10. M. Ramkumar and A. N. Akansu, "A robust oblivious watermarking scheme", in *International Conference on Image Processing - ICIP'2000*, IEEE Signal Processing Society, 2000.
11. N. R. Wagner, "Fingerprinting", in *1983 IEEE Symposium on Security and Privacy*, Oakland CA: IEEE, 1983, pp. 18-22.
12. `http://www.lemuria.org/DeCSS`
13. `http://www.cl.cam.ac.uk/~fapp2/watermarking/benchmark/image_database.html`