IEEE Int. Conf on Multimedia Computing and Systems (ICMCS'99), Florence, Italy, June 1999

Template Based Recovery of Fourier-Based Watermarks Using Log-polar and Log-log Maps

Shelby Pereira, Joseph J. K. Ó Ruanaidh*, Frédéric Deguillaume,
Gabriela Csurka and Thierry Pun
University of Geneva - CUI, 24 rue General Dufour, CH 1211 Geneva 4, Switzerland
Email: {Shelby.Pereira,Frederic.Deguillaume,Gabriela.Csurka,Thierry.Pun}@cui.unige.ch

Abstract

Digital watermarks have been proposed as a method for discouraging illicit copying and distribution of copyrighted material. This paper describes a method for the secure and robust copyright protection of digital images. We present an approach for embedding a digital watermark into an image using the fast Fourier transform. To this watermark is added a template in the Fourier transform domain to render the method robust against rotations and scaling, or aspect ratio changes. We detail a new algorithm based on the log-polar or log-log maps for the accurate and efficient recovery of the template in a rotated and scaled image. We also present results which demonstrate the robustness of the method against some common image processing operations such as compression, rotation, scaling and aspect ratio changes.

1 Introduction

The World Wide Web, digital networks and multimedia afford virtually unprecedented opportunities to pirate copyrighted material. Digital storage and transmission make it trivial to quickly and inexpensively construct exact copies. The idea of using a robust digital watermark to detect and trace copyright violations has therefore stimulated significant interest among artists and publishers.

Many of the current techniques for embedding marks in digital images have been inspired by methods of image coding and compression. Information has been embedded using the Discrete Cosine Transform (DCT) [6, 2], Discrete Fourier Transform (DFT) [7, 4], Wavelets [1], Linear Predictive Coding [5], Fractals [11] and Lapped Orthogonal Transforms [8] as well as in the spatial domain [10, 12]. Recently work has also been done on watermarks which take into account the behavior of human visual system. Objective criteria for measuring the degree to which an image component is significant in watermarking have gradually evolved from being based purely on energy content [6, 2] to statistical [10] and psycho-visual [3] criteria.

Spread spectrum techniques have become a standard method for encoding information in digital image watermarking. It has several advantageous features such as cryptographic security, robustness against noise, and is capable of achieving error free transmission of the watermark near or at the limits set by Shannon's noisy channel coding theorem [9].

The method we propose in the text that follows consists of embedding a watermark in the FFT domain. The watermark is composed of two parts, a template and a spread spectrum signal containing the information or payload. The template contains no information in itself, but is used to detect transformations undergone by the image. Once detected, these transformations are inverted and then the spread spectrum signal is decoded. The spread spectrum signal contains 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. This can be useful for indexing images or even for tracking pornography on the web.

System security is based on proprietary knowledge of the keys (or the seeds for pseudorandom generators) which are required to embed, extract or remove an image watermark. In the case of a public watermarking scheme the key is generally available and may even be contained in publicly available software. In a private watermarking scheme the key is proprietary. From the point of view of embedding watermarks in documents

*Current address: Siemens Corporate Research, 755 College Road East, Princeton, NJ, oruanaidh@scr.siemens.com given the keys or seeds the sequences themselves can be generated with ease. A mark may be embedded or extracted by the key owner which, in our model, is the Copyright Holder. Our system is a private watermarking scheme in which the cryptography aspects are detailed in [4] and will not be addressed here.

Our main contribution lies in the development of an original algorithm for the recovery of rotation, scaling and translation from a log polar map (LPM) of the Fourier transformed stego-image, or aspect ratio from a log-log-map (LLM) from the same this without the need for the original image (oblivious watermarking). In particular, we show how the sampling problems associated with LPMs and LLMs can be overcome and we propose as well a computationally efficient method of calculating the cross correlation between two LPMs or LLMs. After this detection, the transformations can be inverted and the watermark decoded. In the final section we present some results which demonstrate the robustness of the proposed method.

2 Spread Spectrum

In the following, we show how Gold sequences can be used in the watermarking context.

2.1 Encoding the message

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[1] ($\oplus$,{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.

One simple example where one can see this isomorphism at work is in considering the Hamming distance between two binary sequences which is the number of bits by which they differ. It is easy to show that this Hamming distance equals minus the correlation between the two sequences where the bits are replaced by ± 1 as described above.

A method for encoding binary messages which can later be recovered given knowledge of the key used is described here. In all that follows the key is used to determine the initial state of the random number generator. In the case of Gold Codes or m-sequences [9], the key determines the initial state of the registers.

Defining a set of random sequences $\boldsymbol{v}_i$ (shifted m-sequences or Gold Codes) 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 $\boldsymbol{v}_i$.

Decoding is carried out by cross correlating with each of the random sequences $\boldsymbol{v}_i$ in turn. If the correlation is negative then one guesses that a binary one has been sent otherwise one guesses that a binary 0 has been sent.

Clearly, the effectiveness of this scheme depends on the specific choice for the random vectors $\boldsymbol{v}_i$. Maximum length sequences (m-sequences) and Gold Codes were chosen as they combines desirable statistical properties such as uniformly low cross correlation with cryptographic security [9].

This form of spread spectrum is resistant to cropping (providing it is resynchronized), non-linear distortions of amplitude and additive noise. Also, if it has good statistical properties it should be mistaken for noise and go undetected by an eavesdropper. There are however some drawbacks to using direct sequence spread spectrum. Although a spread spectrum signal as described above is extremely resistant to non-linear distortion of its amplitude and additive noise it is also intolerant of timing errors (i.e. getting the starting point for decoding wrong). Synchronization is of the utmost importance during watermark extraction. If watermark extraction is carried out in the presence of the original image then synchronization is relatively trivial. The problem of synchronizing the watermark signal is much more difficult to solve in the present case where there is no original image. If the watermarked image is translated, rotated and scaled then synchronization necessitates a search over a four dimensional parameter space (X-offset, Y-offset, angle of rotation and scaling factor). The search space grows even larger if one takes into account the possibility of shear and a change of aspect ratio. We present in the following fast algorithms for the recovery of the template using log-polar maps and log-log maps which can respectively recover rotation and scale changes or aspect ratio changes.

3 Embedding the Watermark

In order to embed the watermark, we first determine the largest square block that encloses the image. For an image of size (m, n) we choose a blocksize $b = max(m, n)$ an pad the image with 0's to produce a square image during the watermarking. The padded

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

portion is discarded after the watermark has been embedded. The watermark is embedded into the DFT domain between radii $f1$ and $f2$. $f1$ and $f2$ are chosen to occupy a mid-frequency range. We note that the strongest components of the DFT are in the center which contains the low frequencies. Since during the recovery phase the image represents noise, these low frequencies must be avoided. We also avoid the high frequencies since these are the ones most significantly modified during lossy compression such as JPEG. The frequencies $f1$ and $f2$ must be fixed ahead of time since during decoding (in oblivious watermarking) the blocksize b is unknown since the image may have been transformed by cropping or other image processing.

To embed the mark between the chosen radii, we first generate a sequence of points pseudo-randomly as determined by a secret key. Only half the available points in the annulus $\{f1, f2\}$ can be marked since the DFT must be symmetric in order to yield a real image upon inversion. The spread spectrum message generated as described in section 2 is then inserted into these points in the DFT domain. The spectrum is then inverted yielding a spatial domain watermark which is directly added to the image. The image phase is retained since this leads to less artifacts. The strength of the watermark can be set interactively or can be set adaptively as function of the average value and standard deviation of the DFT components of the image lying between $f1$ and $f2$. If the strength is set interactively, the user can examine the artifacts introduced in the image as the strength is increased and finally settle on a strength which is as high as possible while at the same time being relatively invisible.

4 The Template

The template contains no information but is merely a tool used to recover possible transformations in the image. Ultimately, the recovery of the watermark is a two stage process. First we attempt to determine the transformation (if any) undergone by the image, then we invert the transformation and decode the spread spectrum sequence as described in section 2. It is important to note that the watermark detection and decoding is oblivious, that is, we do not require the original image to decode the watermark. This is an important property of the algorithm since if the original were needed, a search in a large database of images would be required to decode the watermark, and this is usually not practical. To the watermark described above is added a template which is used to recover scale as well as rotation, or aspect ratio via the use log-polar maps and log-log-maps.

4.1 Log-polar-mapping

Consider a point $(x, y) \in \Re^2$ and define: $x = e^\mu \cos\theta; y = e^\mu \sin\theta$ where $\mu \in \Re$ and $0 \leq \theta < 2\pi$. One can readily see that for every point (x, y) there is a point (μ, θ) that uniquely corresponds to it.

The new coordinate system has the properties that scaling and rotations are converted to translations. This property makes the template matching problem tractable however the logarithmic sampling of the LPM must be carefully handled in order to recover scaling and rotation information with sufficient accuracy.

4.2 Log-log-mapping

We can likewise derive the properties of the log-log-map. Consider a point $(x, y) \in \Re^2$ and define: $x = e^{\mu_1}; y = e^{\mu_2}$ where $\mu_1, \mu_2 \in \Re$. One can readily see that for every point (x, y) there is a point (μ_1, μ_2) that uniquely corresponds to it. The new coordinate system has the property that changes in aspect ratio are converted to translations. The same sampling problems are incurred with the log-log-map as with the log-polar-map. We consider in section 4.4 how to deal with these problems.

4.3 Embedding the Template

We have found experimentally that using templates of approximately 25 points works best. The points of the template are uniformly distributed in the DFT domain with the low frequencies being excluded. The points are chosen pseudo-randomly as determined by a secret key. Once again the low frequencies are excluded since they contain the bulk of the spectral power and represent noise during the decoding process.

The strength of the template is determined adaptively as well. We find that inserting points at a strength equal to the local average value of DFT points plus one standard deviation yields a good compromise between visibility and robustness during decoding. We note in particular that points in the high frequencies are inserted less strongly since in these regions the average value of the high frequencies is usually lower than the average value of the low frequencies.

4.4 Detecting the Template

We propose in this section a new method whereby the template matching problem is transformed into a point-matching problem over a log-polar map or log-log map.

The algorithm appears below:

1. If the image is rectangular, extract the largest available square from the image.

2. Compute the magnitude of the FFT of the image.

3. Calculate the positions of the local peaks in the DFT using a small window (10 to 14 works well) and store them in a sparse matrix.

4. Compute the corresponding points in log polar space.

5. Compute the positions of the points in log polar space of the known template whose points are generated pseudo-randomly based on a key.

6. Compute the translation offset by exhaustive search which maximizes the numbers of points matched between the known template and the image.

We note that in practice the log-polar map is not completely calculated. Instead, the local peaks are stored in a sparse matrix and their positions in log-polar space are then calculated. Once this is done, we have a point matching problem in which we must determine the x and y offsets which will maximize the match. A number of fast algorithms exist for this type of problem where an optimal match is searched for over a distribution of points where the locations may contain a noise component. However in our case we usually have about a few hundred points to be matched with 25 template points so that an exhaustive search can be done relatively quickly.

The same algorithm can be used to recover changes in aspect ratio as opposed to rotations and scale. The only change is that the log-polar-map is replaced with a log-log-map. However we note that it is impossible to recover a combination of rotation and aspect ratio change with this method since the template will be distorted in both the log-log domain as well as the log-polar domain. Consequently, work still needs to be done to address the general template matching problem.

5 RESULTS

In this section we present some results which demonstrate the robustness of the algorithm to scale changes, rotation changes, cropping, JPEG compressions and combinations thereof. We present the results for the images 512×512 of Lena and Mandrill. The marked images appear in figure 1.

The message “watermark1” was embedded in both images. The following table presents the percent of bits correctly recovered when we attempt to decode the mark after the image has undergone several possible attacks. Table 1 contains the results of the decoding process after the image has been scaled and when the image has been rotated.

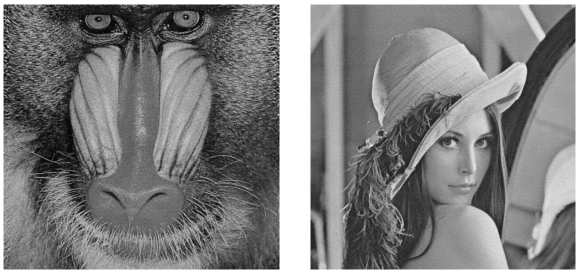

Figure 1: Marked Images of Mandrill and Lena.

Table 1: Percent of Bits Correctly Recovered

Scale	Man.	Lena
0.4	60	66.25
0.5	83.75	98.75
0.7	97.5	100
0.8	100	100
1.5	100	100
1.8	100	100
2	100	100

Rot.	Man.	Lena
15	100	100
30	100	100
45	100	100
130	100	100
150	100	100
200	100	100
240	100	100

The algorithm performs well under scaling and rotation changes. However we note that the decoding becomes inaccurate when the scale change (s) is 0.5 or less. This is simply due to the fact that when we shrink the image we lose information. Nevertheless due to the redundancy in the watermark and the robustness of the spread spectrum, we are still able to decode when the scale has been reduced by $s > 0.5$.

Table 2 contains the results when the watermarked image is compressed with JPEG to various levels and when the image is cropped.

Table 2: Percent of Bits Correctly Recovered

Jpeg	Man.	Lena
75	100	100
50	100	100
30	96	92.5
20	73	67.5

Crop	Man.	Lena
400×400	100	100
350×350	100	98.75
300×300	98	83.75
250×250	91.5	66.25

The results indicate that the method is robust against JPEG up to a quality factor of 30. At this level of compression the image starts to be noticeably degraded and as such it is understandable that the watermark is more difficult to detect. The algorithm also proves very robust to cropping down to 350×350. Below this the loss of information incurred by the cropping renders the decoding of the watermark less reliable.

Table 3 gives the results of combinations of rotations, scaling and cropping. For the cases considered, we see that combinations of scaling, rotation and cropping do not affect the watermark. However when the scaling or cropping is too severe, the watermark is weakened.

Table 3: Percent of Bits Correctly Recovered

combination attack	Man.	Lena
rot=23, sc=0.8	100	100
rot=33,crop 400x400	100	100
sc=0.9, rot=45, crop=400x400	100	98.75

Table 4 contains the results of watermark recovery when the image has been subjected to changes in aspect ratio. The algorithm is fairly robust against such changes, and algorithm starts to break down only when one of the scale changes is below 0.45 or greater than 2. This already corresponds to a significant change in the image. Even at a change in scale of almost 4 along the x axis, 93.25% of the bits are successfully recovered.

Table 4: Percent of Bits Correctly Recovered

aspect ratio	Man.	Lena
x=0.45 y=0.49	100	100
x=0.92 y=0.41	90.6	92
x=1.97 y=0.41	93.75	94.25
x=2.01 y=0.53	100	100
x=3.10 y=0.95	95.31	96.3
x=3.71 y=0.98	92.19	93.25

The algorithm was also tested against printing (Optra S2455 Laser Printer at 175dpi) and rescanning (Epson GT9500 scanner at 300dpi). The watermark was decoded with an accuracy of 98.75% for both the Lena and Mandrill images.

ACKNOWLEDGMENTS

We are grateful to Dr Alexander Herrigel and Digital Copyright Technologies for their work on the security architecture for the digital watermark and for the ongoing collaboration. This work is financed by the Swiss Priority Program on Information and Communication Structures (project Krypict) and by the European Esprit Open Microprocessor Initiative (project JEDI-FIRE). This work is part of the European Patent application EU 978107084.

References

[1] Marco Corvi and Gianluca Nicchiotti. Wavelet based image watermarking for copyright protection. In *The 10th Scandinavian Conference on Image Analysis*, pages 157–163, Lappeenranta, Finland, June 1997.

[2] 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, September 1996.

[3] J. F. Delaigle, C. De Vleeschouwer, and B. Macq. Watermarking algorithm based on a human visual model. *Signal Processing*, 66:319–335, 1998.

[4] Alexander Herrigel, Joe J. K. Ó Ruanaidh, H. Petersen, Shelby Pereira, and Thierry Pun. Secure copyright protection techniques for digital images. In *International Workshop on Information Hiding*, Portland, OR, USA, April 1998.

[5] K. Matsui and K. Tanaka. Video-Steganography: How to secretly embed a signature in a picture. In *IMA Intellectual Property Project Proceedings*, pages 187–206, January 1994.

[6] Joe J. K. Ó Ruanaidh, W. J. Dowling, and F. M. Boland. Watermarking digital images for copyright protection. *IEE Proceedings on Vision, Signal and Image Processing*, 143(4):250–256, 1996.

[7] 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. (Special Issue on Copyright Protection and Control, B. Macq and I. Pitas, eds.).

[8] 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.

[9] 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.

[10] I Pitas. A method for signature casting on digital images. In *Proceedings of the IEEE Int. Conf. on Image Processing ICIP-96*, pages 215–218, Lausanne, Switzerland, September 16-19 1996.

[11] J. Puate and F. Jordan. Using fractal compression scheme to embed a digital signature into an image. In *Proceedings of SPIE Photonics East'96 Symposium*, November 1996.

[12] A. Z. Tirkel, C.F. Osborne, and T.E. Hall. Image and watermark registration. *Signal processing*, 66:373–383, 1998.