Secure Robust Digital Watermarking Using the Lapped Orthogonal Transform

Shelby Pereira, Joseph J. K. Ó Ruanaidh[a] and Thierry Pun

Computer Vision Group, Computer Science Department, University of Geneva,
24 rue du Général Dufour, CH 1211 Geneva 4
Switzerland

ABSTRACT

Digital watermarks have been proposed as a method for discouraging illicit copying and distribution of copyright material. One approach to Transform Domain image watermarking is to divide the image into separate blocks and compute the transform of each block. The watermark is inserted in the transform domain and the inverse transform is then computed. Such an approach is particularly effective against JPEG compression where 8 x 8 blocks are used in conjunction with the DCT. Using small blocks allows the watermark to be embedded adaptively as a function of the luminance and texture. However for small block sizes blocking artifacts are observed when the strength of the watermark is increased. In order to circumvent this problem, we propose a new approach based on Lapped Orthogonal Transforms (LOT) in which the watermark is inserted adaptively into the LOT domain.

Robustness of the watermark to operations such as lossy compression is achieved by using a spread spectrum signal which is added in the LOT domain. The keys used to embed the spread spectrum signal are generated, certified, authenticated and securely distributed using a public key infrastructure containing an electronic copyright office and a certification authority. In addition to the above we propose using an invisible template to reverse the effects of rotation, rescaling and cropping on a watermarked image. This separate invisible template is based on the properties of the Fourier Transform.

Finally, we objectively evaluate the performance of the proposed algorithm in order to demonstrate the robustness of the proposed technique with respect to a number of common image processing including JPEG compression, rotation, scaling and cropping.

Keywords: watermarking, image processing, electronic commerce, template mathing, lapped orthogonal transforms, pattern recognition, spread-spectrum

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 however, 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. The idea of using a robust digital watermark to detect and trace copyright violations has therefore stimulated significant interest among artists and publishers. As a result, digital image watermarking has recently become a very active area of research. Techniques for hiding watermarks have grown steadily more sophisticated and increasingly robust to lossy image compression and standard image processing operations, as well as to cryptographic attack.

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)[1,2] Discrete Fourier Transform magnitude[3,4] and phase,[5] wavelets,[6] Linear Predictive Coding[7] and fractals[8] as well as in the spatial domain in the form of pseudo random noise. The key to making watermarks robust has been the recognition

Send correspondence to Shelby Pereira: shelby.pereira@cui.unige.ch ;(http://cuiwww.unige.ch/∼vision)
[a]Present Address: Siemens Corporate Research
755 College Road East, Princeton
NJ 08540, U.S.
oruanaidh@scr.siemens.com
Thierry Pun: Thierry.Pun@cui.unige.ch

that in order for a watermark to be robust it must be embedded in the *perceptually significant* components of the image.[1,2] The term "perceptually significant" is somewhat subjective but it suggests that a good watermark is one which takes account of the behavior of human visual system.[9] 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[1,2] to statistical[10] and psycho-visual[11,12] criteria.

Digital watermarking is fundamentally a problem in digital communications.[1,13,2] In parallel with the increasing sophistication in modeling and exploiting the properties of the human visual system, there has been a corresponding development in communication techniques. Early methods of encoding watermarks were primitive and consisted of no more than incrementing an image component to encode a binary '1' and decrementing to encode a '0'.[14,1] Tirkel and Osborne[15] were the first to note the applicability of spread spectrum techniques to digital image watermarking. Since then there has been an increasing use of spread spectrum communications in digital watermarking. It has several advantageous features such as cryptographic security,[15,16,2] and is capable of achieving error free transmission of the watermark near or at the limits set by Shannon's noisy channel coding theorem.[1,13]

The method we propose in the following text consists of embedding a watermark in the Lapped Orthogonal Transform (LOT) domain. The motivation for using the LOT as the basis for embedding a watermark is that the DCT may produce blocking artifacts if the strength of the watermark is increased sufficiently. The eye is extremely sensitive to such artifacts and as a consequence the LOT has emerged as a method to deal with this problem. The drawback of the LOT is that it is not robust in itself to cropping, rotation or scaling. Consequently, to the LOT domain watermark we propose the addition of a template in the DFT domain. The template is used to increase the robustness of the watermark with respect to rotations or scale changes. Once detected these transformations can be inverted and the watermark is then recovered from the LOT domain. We note that our method is oblivious, that is the original image is not required for decoding 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.

The watermark contains information such as the owner of the image, a serial number and perhaps flags which indicate the type of content. 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 pseudo-random 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, such as ours, the key is proprietary. From the point of view of embedding watermarks in documents given the keys 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. In this form spread spectrum is a symmetric key cryptosystem. Our system is a private watermarking scheme. The system which provides for the secure exchange of images and keys over the net has been developed and is detailed in[17] and will not be addressed here.

The rest of article is structured as follows. In section 2 we develop the LOT and the associated embedding procedure. In section 3 we review the DFT and discuss the embedding and extraction of the template. In particular,we present log-polar maps and then detail our algorithm for detecting the template in an image which has been rotated, scaled, and/or cropped. Once these changes have been detected the watermark is easily decoded. In section 4, results are presented showing the performance of our algorithm against some common image processing operations such as lossy compression as well as scaling, rotation and cropping. Finally conclusions are presented in section 5.

2. LAPPED ORTHOGONAL TRANSFORMS

The LOT has become a standard tool in compression, speech coding and communications applications. Its main advantage over other transforms such as the DCT, the Handamard transform and the Hartley transform is that it overcomes the problem of blocking artifacts. These artifacts arise most notably in JPEG compression at low quality factors, and are caused by the fact that the image is split into blocks which are independently transformed. The transformed blocks are compressed in the DCT domain and then inverted. Upon inversion, there is no reason to expect continuity at the block boundaries and consequently at low quality factors, we observe blocking artefacts.

A detailed discussion and derivation of Lapped Orthogonal Transforms is given by Malvar.[18] Here we present only the essential elements which will be used in our watermarking scheme.

2.1. Definition

We begin by presenting the LOT in 1D, the extension to 2D images is straightforward. The main idea of the LOT is to map 2 or more blocks from the spatial domain to one block in the transform domain. Here we limit ourselves to the case where 2 blocks are mapped to one. That is we have $L = 2M$ where L is the size of the block in spatial domain and M is the size of the domain in the transform domain. For a signal of length $n = NL$ where N is an integer, the transformation matrix between the signal and its transform is:

$$\tilde{\boldsymbol{P}} = \begin{bmatrix} \boldsymbol{P}_a & & & & \\ & \boldsymbol{P} & & & \\ & & \ddots & & \\ & & & \boldsymbol{P} & \\ & & & & \boldsymbol{P}_b \end{bmatrix}; \boldsymbol{P} = \frac{1}{2}\begin{bmatrix} \boldsymbol{D}_e - \boldsymbol{D}_o & \boldsymbol{D}_e - \boldsymbol{D}_o \\ \boldsymbol{J}(\boldsymbol{D}_e - \boldsymbol{D}_o) & -\boldsymbol{J}(\boldsymbol{D}_e - \boldsymbol{D}_o) \end{bmatrix} \tag{1}$$

where:

$$\begin{aligned} \boldsymbol{P}_a &= \tfrac{1}{2}\begin{bmatrix} 2\boldsymbol{D}_{e2} & 2\boldsymbol{D}_{e2} \\ \boldsymbol{J}(\boldsymbol{D}_e - \boldsymbol{D}_o) & -\boldsymbol{J}(\boldsymbol{D}_e - \boldsymbol{D}_o) \end{bmatrix}; & D_{e2} &= \begin{bmatrix} \boldsymbol{H}_e \\ \boldsymbol{J}\boldsymbol{H}_e \end{bmatrix} \\ \boldsymbol{P}_b &= \tfrac{1}{2}\begin{bmatrix} \boldsymbol{D}_e - \boldsymbol{D}_o & \boldsymbol{D}_e - \boldsymbol{D}_o \\ 2\boldsymbol{D}_{e3} & -2\boldsymbol{D}_{e3} \end{bmatrix}; & D_{e3} &= \begin{bmatrix} \boldsymbol{H}_{e2} \\ \boldsymbol{J}\boldsymbol{H}_{e2} \end{bmatrix} \end{aligned} \tag{2}$$

$$\boldsymbol{J} = \begin{bmatrix} & & & 1 \\ & & \cdot & \\ & \cdot & & \\ 1 & & & \end{bmatrix} \tag{3}$$

$\boldsymbol{D}_e$ and $\boldsymbol{D}_o$ contain the even and odd coefficients of the DCT and $\boldsymbol{H}_e$ and $\boldsymbol{H}_{e2}$ contain the first and second half of the even DCT coefficients respectively. The matrix $\tilde{\boldsymbol{P}}$ contains $N-2$ sub-matrices $\boldsymbol{P}$. The matrix $\boldsymbol{P}$ is $L \times M$ so that in the matrix $\tilde{\boldsymbol{P}}$ the upper M rows of a given sub-matrix $\boldsymbol{P}$ line up with the lower M rows of the sub-matrix $\boldsymbol{P}$ diagonally adjacent to it.

With these definitions we have that for a given signal x, its LOT is given by $\tilde{x} = \tilde{\boldsymbol{P}}^t x$. It can be verified directly that $\tilde{\boldsymbol{P}}\tilde{\boldsymbol{P}}^t = I$ so that from the transformed vector $\tilde{x}$ we can recover the original signal exactly by applying the inversion transform $\tilde{\boldsymbol{P}}$. The extension to two dimensions is straightforward: the image rows are first transformed yielding an intermediate image for which the columns are transformed.

2.2. Embedding the Watermark

We describe here how to embed a 32 bit watermark in an image. The watermarking procedure is composed of five steps. (1) Divide the image into 128×128 pixel blocks and compute the LOT of each block; if the image is in color, only the luminance component is marked; (2) convert the 32 bit watermark into a spread spectrum (SS) sequence; (3) choose coefficients pseudo-randomly in the LOT domain; (4) add the SS sequence to selected coefficients; (5) perform the inverse LOT. These steps are detailed below.

We need to divide the image into blocks in order to resist cropping as will become evident in section 3.3. If the image size is not a multiple of 128, the remaining parts are left unmarked. The 128×128 blocks are further subdivided into sub-blocks of size 16×16 ($M = 16$,$L = 32$). The motivation for using small sub-blocks is that it will eventually allow us to produce watermarking schemes which are locally adaptive. In other words we wish to modulate the strength of the watermark in a given block as a function of local image characateristics as will be made clear in section 2.3.

After computing the LOT, we must transform the 32 bit binary watermark into a spread spectrum (SS) sequence. Suppose we are given a message which is represented in binary form as $\boldsymbol{m} = (m_1, m_2 ... m_M)$ where $m_i \in \{0, 1\}$ and M is the number of bits in the message. The binary form of the message $\boldsymbol{m}$ is then transformed to obtain the vector $\boldsymbol{b} = (b_1, b_2, ... b_M)$, with $b_i \in \{1, -1\}$ by exploiting the basic isomorphism between the group ($\oplus$ (0,1)) and the group (*(1,-1)) as illustrated by table 2.2 below. The mapping 1→-1 and 0→1 is an extremely important step because it essentially enables us to replace the exclusive-OR operator with multiplication. This is useful when decoding real

*	1	-1
1	1	-1
-1	-1	1

$\oplus$	0	1
0	0	1
1	1	0

Table 1. Isomorphism between groups ($\oplus$ (0,1)) and the group (*(1,-1))

valued sequences such as digital watermarks. 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.

We define a set of random sequences $\boldsymbol{v}_i$ corresponding to the bits b_i. The encoded message is given by:

$$\boldsymbol{W} = \sum_{i=1}^{M} b_i \boldsymbol{v}_i \tag{4}$$

or using matrix notations:

$$\boldsymbol{m} = \boldsymbol{G}\boldsymbol{b} \tag{5}$$

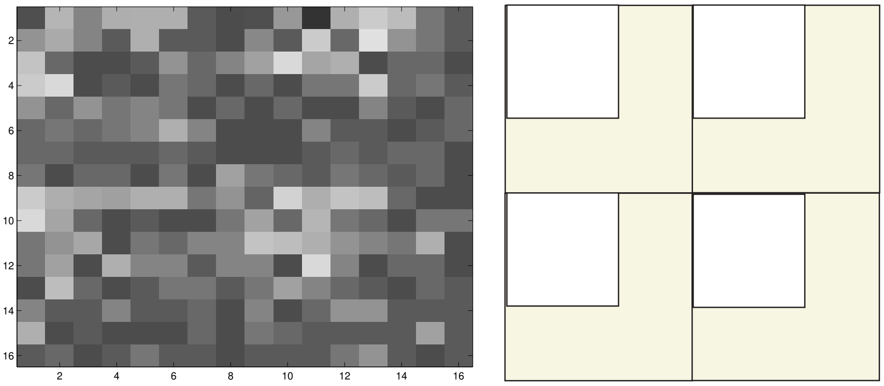
where $\boldsymbol{b}$ is a $M \times 1$ vector of bits (in ± 1 form), $\boldsymbol{m}$ is a $N \times 1$ vector and $\boldsymbol{G}$ is an $N \times M$ matrix such that the i^{th} column is the vector $\boldsymbol{v}_i$.

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 $\boldsymbol{v}_i$. Ideally the vectors should be as well separated as possible to get the maximum descrimination between the bits. A good spread spectrum sequence is one which combines desirable statistical properties such as uniformly low cross correlation with cryptographic security. We choose to use M-sequences since it is known that these sequences have good cross correlation properties. Since we have $M = 16$ we have a maximum of $M^2 = 256$ points which can be marked per

(a) Lena (b) LOT of Lena

Figure 1. (a) Lena and (b) LOT of Lena (log corrected for display).

sub-block. In order for the watermark to be oblivious, we must avoid marking the high energy components in the LOT domain. Figure 1 contains the original Lena image along with its LOT. One block of the LOT appears in figure 2a. We note that the dominant components are concentrated in the four upper left corners of the sub-blocks. This results directly from the fact that we transform two spatial domain blocks to one time domain block. Consequently we choose to embed the mark in the regions of each block as indicated by the shaded regions in figure 2b. Since



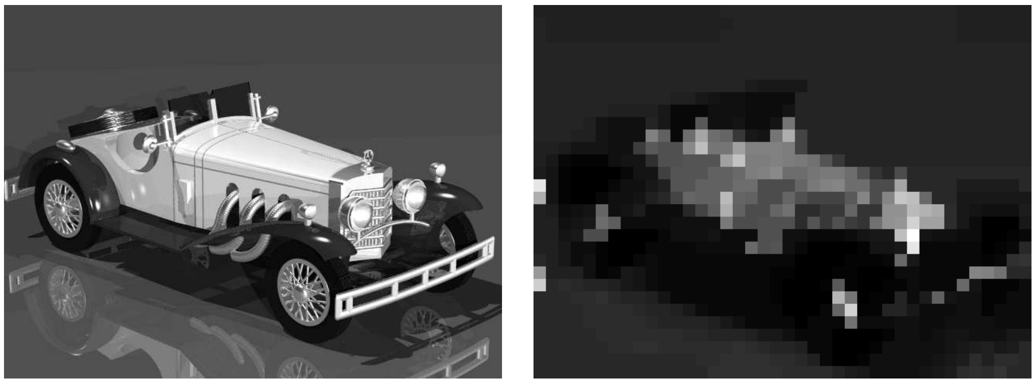
(a) One block in LOT domain (b) Region for embedding

Figure 2. (a) One block in LOT domain and (b) region used for embedding

M-sequences must be of length $2^n - 1$ with n an integer, we choose a length of 63 so that we can fit an entire sequence into one block without marking the high energy components. Once the SS sequence has been genereated we pseudo-randomly select points in the LOT domain. The SS sequence is then added to each block and the inverse LOT calculated.

2.3. Adaptive Watermarking

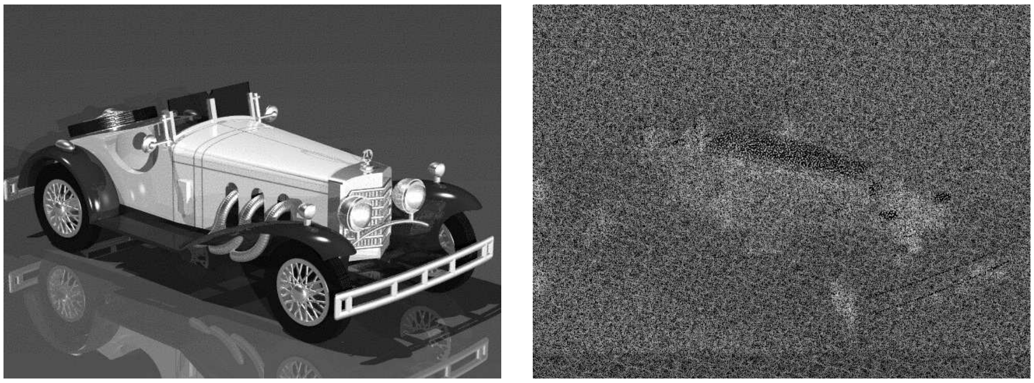
As was mentioned in the previous section, the main motivation for using small blocks is that the resulting algorithm can be rendered locally adaptive. In particular, we would like to embed the watermark more strongly in regions where the luminance is high or in regions which are highly textured. We adopt this strategy since it is well known that the eye is less sensitive to changes in bright regions and in textured regions. Consequently after computing the SS sequence,we multiply the sequence by a different constant for each block as determined by the following strategy: (1) Calculate local mean for the 16×16 block; (2) Calculate a function of the local texture in the 16×16 block, here we use the number of edges, where the edges are computed by a 3×3 Sobel filter; (3) define $maskconstant = k1 * mean + k2 * texture$; (4) for a given block we watermark using: $strength = basestrength + maskconstant$.



(a) Benz (b) Mask of Benz

Figure 3. (a) Benz (Image courtesy of F. Petticolas) and (b) Mask of Benz.

Using this strategy, for the image in figure 3a, we obtain the mask in figure 3b. The watermarked image appears in figure 4 along with the watermark (difference between original and watermarked image).



(a) Watermarked Benz image (b) Watermark

Figure 4. (a) Watermarked Benz image and (b) Watermark.

3. THE TEMPLATE

The LOT in itself is not resistant to cropping, rotations or scale changes. Consequently, in order to improve the robustness of the watermark, we propose adding a template in the DFT domain. The DFT domain is suitable since linear transformations can easily be detected as will be seen in section 3.1. 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 2.2.

We have found experimentally that using templates of approximately 25 points work best. The points of the template are uniformly distributed in the DFT domain and their locations are chosen pseudo-randomly as determined by a secret key. The low frequencies are excluded since they contain the bulk of the power of the spectrum and represent noise during the decoding process. The template is embedded in blocks of 128×128. At the edges of the image, we pad with zeros if necessary, embed the template and truncate the zone in which the zeros were added.

The points of the template lie in a band between the normalized frequencies 0.3 and 0.4. The strenght of the points is equal to the average in the band plus 4 standard deviations and the phase of the points are set to zero. This insures that during the detection process we can match peaks as detailed in section 3.3.

3.1. General Properties of the Fourier Transform

Before considering the problem of recovering a watermark from a rotated and scaled image, we study the effect of an arbitrary linear transform on the spectrum of an image. Since our template is embedded in the DFT domain, if we can determine the transformation $\boldsymbol{T}$ undergone by the image, it is possible to perform the inverse transformation and then recover the watermark from the LOT domain.

The Discrete Fourier Transform (DFT) is defined as follows:

$$F(k_1, k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f(x_1, x_2) e^{-j2\pi x_1 k_1/N_1 - j2\pi x_2 k_2/N_2} \tag{6}$$

When we have square blocks the kernel of the DFT contains a term of the form:

$$x_1 k_1 + x_2 k_2 = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \tag{7}$$

If we compute a linear transform on the spatial coordinates:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \to \boldsymbol{T} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{8}$$

then one can see that the value of the DFT will not change if:

$$\begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \to \left(\boldsymbol{T}^{-1}\right)^T \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \tag{9}$$

The matrix $\boldsymbol{T}$ is an arbitrary matrix which can be a composition of scale changes, rotations, and/or skews. In the following we will discuss how to recover watermarks when the matrix $\boldsymbol{T}$ is a composition of scale and rotation changes.

Another important property of the DFT is its translation invariance. In fact, shifts in the spatial domain cause a linear shift in the phase component.

$$F(k_1, k_2)\exp\left[-j(ak_1 + bk_2)\right] \leftrightarrow f(x_1 + a, x_2 + b) \tag{10}$$

3.2. Log-polar-mapping

In this section we show how scaling and rotation changes can be recovered from a template. We first consider the properties of log-polar maps which will be used in the template matching algorithm. We then present a new algorithm for detecting the template in spite of the sampling problems associated with the logarithmic scale.

Consider a point $(x, y) \in \Re^2$ and define:

$$\begin{aligned} x &= e^{\mu}\cos\theta \\ y &= e^{\mu}\sin\theta \end{aligned} \tag{11}$$

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 following properties:

Scaling is converted to a translation.

$$(\rho x, \rho y) \leftrightarrow (\mu + \log\rho, \theta) \tag{12}$$

Rotation is converted to a translation.

$$\begin{aligned} &(x\cos(\delta) - y\sin(\delta), x\sin(\delta) + y\cos(\delta)) \\ &\quad\leftrightarrow (\mu, \theta + \delta) \end{aligned} \tag{13}$$

3.3. Detecting the Template and Decoding the Watermark

The problem of template matching using log-polar-maps has been recently addressed by Bonnmaser[19] where the Fast exponential chirp transform is used. Here we propose an alternative whereby the template matching problem is transformed into a point-matching problem over a log-polar map. The template matching process is divided into two parts. In the first part, we extract the rotation and scale changes and then invert them. In the second part, we calculate the offset to the nearest 128×128 block. This second step is necessary since the LOT is not translation invariant. Consequently, correctly locating the starting point is essential to the recovery of the watermark.

The rotation and scale compensation is performed as follows: (1) if the image is rectangular, extract the largest available square from the image; (2) compute the magnitude of the DFT of the windowed image; (3) calculate the positions of the local peaks in the filtered 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 locations were generated pseudo-randomly based on a key; (6) exhaustively search the space of possible matches between the known template and the extracted peaks and thereby deduce the optimal offset; (7) use the offset to compute the scale and rotation change using equations 12 and 13; (8) invert the rotation and scale.

In the second part of the template matching, we calculate the offset to the starting point of a block by means of a fast cross-correlation as described by Secilla.[20] We briefly review the algorithm below which directly follows from the fact that the translation information is contained in the phase. Since the template is encoded with a known phase 0, we use the template points to compute the offset: (1) generate the known 128×128 template in the DFT domain; (2) calculate the DFT of a 128×128 block of the image; (3) generate a new DFT composed of the magnitude of the DFT of the template and the phase of the DFT of the image; (4) invert the constructed DFT; (5) the location of the maximum in the spatial domain is the offset value.

Once the offset has been calculated, we calculate the LOT starting at the offset point. For each transform domain block, we recover the SS sequence embedded in the block. These are all added together since we assume that the noise has a mean equal to 0. The resulting SS sequence is decoded as described in 2.2.

4. 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 of Lena and Mandrill both of size 512×512. The marked images appear in figure 5.

Figure 5. Marked Images of Lena and Mandrill.

The 32 bit message "1234" 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 2 contains the results of the decoding process after the image has been scaled or rotated.

Table 2. Percent of Bits Correctly Recovered During Decoding

Scale Change	Mandrill	Lena
0.6	60	66.25
0.65	83.75	98.75
0.75	97.5	100
0.8	100	100
1.5	100	100
1.8	100	100
2	100	100

Rotation	Mandrill	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 is 0.65 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 been reduced by 0.65.

Table 3 contains the results when the watermarked image is compressed with JPEG to various levels and when the image is cropped.

The results indicate that the method is robust against JPEG down 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

Table 3. Percent of Bits Correctly Recovered During Decoding

JPEG factor	Mandrill	Lena
75	100	100
50	100	100
30	96	92.5
20	73	67.5

Cropping	Mandrill	Lena
400x400	100	100
350x350	100	98.75
300x300	98	83.75
250x250	91.5	66.25

difficult to detect. The algorithm also proves robust to cropping down to a level of 350x350. Below this the loss of information incurred by the cropping renders the decoding of the watermark much less reliable.

Table 4 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 4. Percent of Bits Correctly Recovered During Decoding

combination attack	Mandrill	Lena
rot=23, scale=0.8	100	100
rot=33,cropping 400x400	100	100
scale=0.9, rot=45, crop=400x400	100	98.75

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.

5. CONCLUSION

We have presented here an approach for image watermarking in which the watermark is embedded in the LOT domain. Our main contribution lies in the development of a fast algorithm for the recovery of scale and rotation from the log-polar-map. The algorithm we present is efficient and robust. We have also presented results which demonstrate the viability of the method in practice. Current work is being directed to solve the general template matching problem in the context of watermarking so that arbitrary transformations can be detected from the template embedded in the DFT domain.

ACKNOWLEDGMENTS

We are grateful to Dr Alexander Herrigel and Digital Copyright Technologies Switzerland 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 EU978107084. Thanks also to Gabriela Csurka and Frederic Deguillaume for their very useful insights.

REFERENCES

1. J. J. K. Ó Ruanaidh, W. J. Dowling, and F. M. Boland, "Watermarking digital images for copyright protection," *IEE Proceedings on Vision, Image and Signal Processing* **143**, pp. 250–256, August 1996. Invited paper, based on the paper of the same title at the IEE Conference on Image Processing and Its Applications, Edinburgh, July 1995.
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*, pp. 243–246, (Lausanne, Switzerland), September 16-19 1996.
3. J. Oruanaidh and T. Pun, "Rotation, scale and translation invariant spread spectrum digital image watermarking," *Signal Processing* **66**(3), pp. 303–317, 1998.

4. J. Oruanaidh and S. Pereira, "A secure robust digital image watermark," *Electronic Imaging: Processing, Printing and Publishing in Colour* , May 1998.
5. J. J. K. Ó Ruanaidh, W. J. Dowling, and F. M. Boland, "Phase watermarking of digital images," in *Proceedings of the IEEE Int. Conf. on Image Processing ICIP-96*, pp. 239–242, (Lausanne, Switzerland), September 16-19 1996.
6. M. Corvi and G. Nicchiotti, "Wavelet-based image watermarking for copyright protection," in *The 10th Scandinavian Conference on Image Analysis*, June 1997.
7. K. Matsui and K. Tanaka, "Video-Steganography: How to secretly embed a signature in a picture," in *IMA Intellectual Property Project Proceedings*, pp. 187–206, January 1994.
8. 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.
9. J. F. Delaigle, C. De Vleeschouwer, and B. Macq, "Watermarking algorithm based on a human visual model," *Signal Processing* **66**, May 1998.
10. I. Pitas, "A method for signature casting on digital images," in *Proceedings of the IEEE Int. Conf. on Image Processing ICIP-96*, pp. 215–218, (Lausanne, Switzerland), September 16-19 1996.
11. M. D. Swanson, B. Zhu, and A. Tewfik, "Transparent robust image watermarking," in *Proceedings of the IEEE Int. Conf. on Image Processing ICIP-96*, pp. 211–214, (Lausanne, Switzerland), September 16-19 1996.
12. J. Delaigle, C. De Vleeschouwer, and B. Macq, "Digital Watermarking," in *Conference 2659 - Optical Security and Counterfeit Deterrence Techniques*, SPIE Electronic Imaging : Science and Technology, (San Jose), Feb. 1996. pp. 99-110.
13. J. Smith and B. Comiskey, "Modulation and information hiding in images," in *Proceedings of the First International Workshop in Information Hiding*, R. Anderson, ed., Lecture Notes in Computer Science, pp. 207–226, Springer Verlag, (Cambridge, UK), May/June 1996.
14. G. Caronni, "Assuring Ownership Rights for Digital Images," in *Reliable IT Systems VIS '95*, H. H. Brueggemann and W. Gerhardt-Haeckl, eds., Vieweg Publishing Company, Germany, 1995.
15. 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*, pp. 666–672, (Macquarie University, Sydney), December 1993.
16. A. Z. Tirkel, R. G. van Schyndel, and C. F. Osborne, "A two-dimensional digital watermark," in *Dicta'95*, pp. 378–383, (University of Queensland, Brisbane), December 6-8 1995.
17. A. Herrigel, J. Ó Ruanaidh, H. Petersen, T. Pun, and S. Pereira, "Copyright protection for digital images based on asymmetric cryptographic and image authentication techniques," in *Electronic Image Capture and Publishing EUROPTO-98*, (Zurich, Switzerland), 1998.
18. H. S. Malvar, *Signal Processing with Lapped Orthogonal Transforms*, Artech House Inc., 1992.
19. G. Bonmassar and E. Schwartz, "Space-variant fourier analysis: The exponential chirp transform," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, October 1997.
20. J. P. Secilla and N. Garcia, "Template location in noisy pictures," *Signal Processing* **14**, pp. 347–361, 1988.