Robust template matching for affine resistant image watermarks

Pereira, Shelby; Pun, Thierry

How to cite

PEREIRA, Shelby, PUN, Thierry. Robust template matching for affine resistant image watermarks. In: IEEE transactions on image processing, 2000, vol. 9, n° 6, p. 1123–1129. doi: 10.1109/83.846253

This publication URL: https://archive-ouverte.unige.ch/unige:47477
Publication DOI: 10.1109/83.846253

Robust Template Matching for Affine Resistant Image Watermarks

Shelby Pereira and Thierry Pun

University of Geneva - CUI, 24 rue General Dufour, CH 1211 Geneva 4, Switzerland

Email: {Shelby.Pereira,Thierry.Pun}@cui.unige.ch

April 22, 1999

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 Fourier transform. To this watermark is added a template in the Fourier transform domain to render the method robust against general linear transformations. We detail a new algorithm based on polar maps for the accurate and efficient recovery of the template in an image which has undergone a general affine transformation. 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.

EDICS number=5-AUTH Authentication and Watermarking

Correspondance should be sent to Shelby Pereira

Shelby.Pereira@cui.unige.ch

phone: 0 11 41 22 705 7631

FAX: 0 11 41 22 705 7780

address:University of Geneva

CUI, 24 rue General Dufour

CH 1211 Geneva 4

Switzerland

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. Consequently, the idea of using a robust digital watermark to detect and trace copyright violations has therefore stimulated significant interest among artists and publishers.

In order for a watermark to be useful it must be robust against a variety of possible attacks by pirates. These include robustness against compression such as JPEG, scaling and aspect ratio changes, rotation, cropping, row and column removal, addition of noise, filtering, cryptographic and statistical attacks, as well as insertion of other watermarks. A discussion of possible attacks is given by Petitcolas and Craver.[15,3] Watermarking methods have become increasingly more robust against the above mentioned attacks.

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),[8,2] Wavelets,[1] Linear Predictive Coding,[10] and Fractals[17] as well as in the spatial domain .[16,21] While these methods perform well against compression, they lack robustness to geometric transformations. Consequently methods have emerged which exploit the properties of the Discrete Fourier Transform (DFT) to achieve robustness against rotation and scaling. The DFT methods can be divided into two classes, those based on invariance [12,7] and those which embed a template into the image which is searched for during the detection of the watermark and yields information about the transformation undergone by the image.[13,18] However both these methods exploit the properties of log-polar-maps (LPM) and can only be used to detect changes of rotation and scale. Similarly the log-log-map (LLM)[4] has also been proposed as a means of detecting changes in aspect ratio. However, once again general transformations cannot be recovered.

The method we propose in the text that follows consists of embedding a watermark in the DFT domain. The watermark is composed of two parts, a template and a spread spectrum message 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 payload 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 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 the key is proprietary. From the point of view of embedding watermarks in documents 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 by Herrigel[7] and will not be addressed here. In what follows, we limit ourselves to the image processing aspects of the problem.

The main contribution of this article lies in the development of a method for recovering a watermark from an image which has undergone a general affine transformation. Unlike algorithms which use LPMs and LLMs to recover rotation, scale and aspect ratio information, we propose introducing structure into the template which is exploited at the detection stage to reduce the search space. The proposed method is evaluated relative to the benchmark series of tests proposed by Kutter and Petitcolas[9] and implemented in the software package Stirmark3.[14] The algorithm performs very well relative to the extensive series of tests implemented in the benchmark.

The rest of this paper is structured as follows. In section 2 we describe the embedding approach. Section 3 describes the extraction algorithm. In Section 4, we present our results. Finally, section 5 contains our conclusions.

2. EMBEDDING

In this section we describe the embedding approach. First we show how the message is encoded. Secondly we review some key properties of the DFT before demonstrating how the encoded message is inserted in this domain. We conclude this section by showing how the template is also embedded in the DFT domain.

2.1. Encoding the message

In image watermarking, we are given a message to be embedded which can be 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. In realistic applications

M is roughly 60 bits which contain the necessary copyright information as well as flags which can be used to indicate the type of content in the image. In our scheme, the binary message is first coded using the well known BCH codes[19] to produce the message $\boldsymbol{m}_c$ of length $M_c = 72$. We then apply the mapping $0 \to -1$ and $1 \to 1$ to produce the bipolar signal $\tilde{\boldsymbol{m}}_c = (\tilde{m}_{c1}...\tilde{m}_{cM_c})$ which can then be embedded as described in section 2.3.

2.2. The DFT and its Properties

2.2.1. Definition

Let the image be a real valued function $f(x_1, x_2)$ defined on an integer-valued Cartesian grid $0 \leq x_1 < N_1, 0 \leq x_2 < N_2$.

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

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

The inverse transform is

$$f(x_1, x_2) = \frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} F(k_1, k_2) e^{j2\pi k_1 x_1/N_1 + j2\pi k_2 x_2/N_2} \tag{2}$$

The DFT of a real image is generally complex valued. This leads to magnitude and phase representation for the image:

$$A(k_1, k_2) = |F(k_1, k_2)| \tag{3}$$

$$\Phi(k_1, k_2) = \angle F(k_1, k_2) \tag{4}$$

2.2.2. General Properties of the Fourier Transform

It is instructive to study the effect of an arbitrary linear transform on the spectrum of an image.

Once $N_1 = N_2$ (i.e. square blocks) the kernel of the DFT contains a term of the form:

$$x_1 k_1 + x_2 k_2 = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \tag{5}$$

If we compute a linear transform on the spatial coordinates:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \boldsymbol{T} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{6}$$

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

$$\begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \rightarrow \left(\boldsymbol{T}^{-1}\right)^T \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \tag{7}$$

Since our watermarks are embedded in the DFT domain, if we can determine the transformation $\boldsymbol{T}$ undergone by the image in the spatial domain, it will be possible to compensate for this transformation in the DFT domain and thereby recover the watermark. The matrix $\boldsymbol{T}$ is an arbitrary matrix which can be a composition of scale changes, rotations, and/or skews. In section 3.1 we will discuss how to recover watermarks when an arbitrary matrix $\boldsymbol{T}$ is applied to the image.

2.2.3. DFT: Translation

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.

$$F(k_1, k_2) \exp\left[-j(ak_1 + bk_2)\right] \leftrightarrow f(x_1 + a, x_2 + b) \tag{8}$$

From equation 8 of the Fourier transform it is clear that spatial shifts affect only the phase representation of an image. This leads to the well known result that the magnitude of the Fourier transform is invariant to translations in the spatial domain. This property leads directly to the fact that the watermark is robust against cropping.

2.3. Embedding the Watermark

When working with color images, we first extract the luminance component and then rescale the RGB components accordingly. In order to embed the watermark for an image of size (m, n), we first pad the image with zeros so that the resulting size is 1024×1024. If the image is larger than 1024×1024 then the image is divided into 1024×1024 blocks and the watermark is calculated for each block. The watermark

*The DFT will be invariant except for a scaling factor which depends on the Jacobian of Transformation, namely the determinant of the transformation matrix $\boldsymbol{T}$.

is embedded into the DFT domain between radii f_{w1} and f_{w2} where f_{w1} and f_{w2} 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 as illustrated in figure 1. 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.

To embed the mark between the chosen radii, we first generate a sequence of points $(x_1, y_1)...(x_{M_c}, y_{M_c})$ pseudo-randomly as determined by a secret key. Here, x_i, y_i are integers such that $f_{w1} < \sqrt{x_i^2 + y_i^2} < f_{w2}$. We note that only half the available points in the annulus $\{f_{w1}, f_{w2}\}$ can be marked since the DFT must be symmetric in order to yield a real image upon inversion. In what follows we work in the upper half plane and assume that the corresponding modifications are made in the lower half plane $(\pm x_i, y_i)$ to fulfill the symmetry constraints.

Since the magnitude of the DFT is positive valued, in order to encode the bipolar message $\tilde{M}_c$, we adopt the following differential encoding scheme. For each message bit $\tilde{m}_{c_i}$ we modify the points (x_i, y_i) and $(y_i, -x_i)$ such that $k_w \tilde{m}_{c_i} = (x_i, y_i) - (y_i, -x_i)$. In other words 2 points 90° apart are modified such that the difference is equal to the desired message value. The parameter k_w is the strength of the watermark. The watermark strength 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 f_{w1} and f_{w2}. 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 leaving the watermark relatively invisible.

2.4. Embedding the Template

The template contains no information but is merely a tool used to recover possible transformations in the image. In previous work the template consisted of a random arrangement of peaks in the FFT domain.[13] Here we propose introducing structure into the template which can be exploited during the recovery phase to make a search for a general linear transformation tractable.

We have found experimentally that using templates of approximately 14 points works best. The points of the template are distributed uniformly along 2 lines (7 points per line) in the DFT domain at angles θ_1 and θ_2 with radii varying between f_{t1} and f_{t2}. The angles (θ_i) and radii (r_{ij}) are chosen pseudo-randomly

as determined by a secret key. 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 two standard deviations 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.

3. DECODING

The watermark extraction process is divided into two phases. First we have the template detection phase and then we decode the watermark if the template has been detected.

3.1. Template Detection

The template detection process involves several steps which we detail below. The main idea is to exploit the fact that the template points have been embedded along two lines which go through the origin as described in section 2.4. The main observation is that an image which has undergone a linear transformation will have undergone the inverse linear transformation in the DFT domain. Furthermore, for a linear transformation, a line going through the origin will be transformed into a corresponding line going through the origin. The radii of the new points $\boldsymbol{r}'$ will be related to the radii of the old points $\boldsymbol{r}$ by $\boldsymbol{r}' = K\boldsymbol{r}$ for some constant K. These observations will be exploited to yield the fast algorithm below.

1. Apply a Bartlett window to the spatial domain image $\boldsymbol{I}$ to produce $\boldsymbol{I}_w$.
2. Calculate the FFT of the image padded to 1024×1024.
3. Extract the positions of all the local peaks (p_{xi}, p_{yi}) in the image.
4. Map the positions of the peaks to polar coordinates i.e. $(p_{xi}, p_{yi}) \rightarrow (r_{Ii}, \theta_{Ii})$.
5. Sort the peaks by angle and divide into N_b equally spaced bins by angle.
6. For both lines in the template perform the following.

   For each of the N_b equally spaced bins search for a K where $K_{min} < K < K_{max}$ such that at least N_m points match between the points $\boldsymbol{r}_{Ii}$ which are the radial coordinates of the points in bin i where $i \in 1 \ldots N_b$ and $\boldsymbol{r}_{Tj}$ which are the radial coordinates of the template along line j where $j \in 1, 2$. Here

two points match if for two given radial coordinates r_{Ii} and r_{Tj} we have $|r_{Ii} - Kr_{Tj}| < thresh$. If at least N_m points match, we store the set of matched points. We note that the number of sets of matched points is equal to the number of lines for which we have at least N_m matched radii. We also note that the value of K is searched over all $r_{Ii}/r_{Tj} = K$ such that K falls in the allowable range.

7. For all combinations of sets of matched points choosing one set from those corresponding to template line 1 and a second corresponding to template line 2 (corresponding to matches against lines embedded at θ_1 and θ_2 such that the angle between points in 2 given lists is within θ_{diff} of the difference between θ_1 and θ_2), calculate the linear transformation $\boldsymbol{A}$ such that the mean square estimation error in equation 9 is minimized.

$$mse = \frac{1}{nummatches} \| \boldsymbol{A} \begin{bmatrix} x'_{11} & y'_{11} \\ \vdots & \vdots \\ x'_{1l} & y'_{1l} \\ x'_{21} & y'_{21} \\ \vdots & \vdots \\ x'_{2l} & y'_{2l} \end{bmatrix}^T - \begin{bmatrix} x_{11} & y_{11} \\ \vdots & \vdots \\ x_{1l} & y_{1l} \\ x_{21} & y_{21} \\ \vdots & \vdots \\ x_{2l} & y_{2l} \end{bmatrix}^T \|^2 \tag{9}$$

   We note that $\boldsymbol{A}$ is a 2×2 linear transformation matrix so that we understand the notation $\|.\|$ to mean the sum of the magnitude of the two rows of the error matrix. The rows contain the errors in estimating the $\boldsymbol{x}$ and $\boldsymbol{y}$ from the known template positions $\boldsymbol{x}'$ and $\boldsymbol{y}'$ after applying the transformation $\boldsymbol{A}$.

8. Repeat the previous step adding 180° to the angles in the sets of matched points corresponding to line 1 of the template (either line can be used).

9. Choose the $\boldsymbol{A}$ which minimizes the error.

10. If the minimized error is less than the detection threshold T_d, we conclude that the watermark is detected and proceed to decoding. Otherwise we conclude that no watermark was embedded in the image.

Some observations are necessary. Firstly, in step 1 the Bartlett window is applied to eliminate artifacts associated with the implicit assumption of periodicity in the image in the calculation of the DFT. Secondly,

rather than adopting an expensive search for lines we use the fast algorithm in steps 2-5 which proves sufficiently robust for our application. Placing the points in bins according to angle is roughly equivalent to searching for points which fall on a line going through the origin. We note that this is a special case of the well known Hough transform.

Thirdly, step 8 is necessary to resolve ambiguities associated with the fact that we only use the upper half plane. If we consider figure 2 it is clear that after a large rotation the angle between the two lines is not preserved if we consider only the upper half plane ($\theta_1 \rightarrow \theta_2$). Consequently we must add 180° to the angles of the points in one line to resolve the ambiguity. We note that in practice only one case will be considered in equation 9 since the other case will be eliminated since the difference between angles will be greater than θ_{diff}.

Finally we note that step 10 contains a criterion for asserting the presence of a watermark. Consequently the template serves the dual purpose of recovering geometrical transformations and asserting the presence of a watermark even if the watermark itself may be falsely decoded. Some alternate schemes consist of using cross-correlation as in[20,22,5] or Bayesian models as in.[11] The advantage of using the template is that it is much more robust than the watermark itself since we concentrate a significant amount of energy into a few points in the FFT. However, work still needs to be done in the development of a statistical model which will establish some measure of confidence in the match. At present the detection threshold is set empirically to minimize false positives. In practice these occur extremely rarely since firstly we require that a minimum of N_m points match in a given line, secondly we require that change in angle by less than θ_{diff} and thirdly we insist that the estimated linear transformation matrix yields an estimation error which is no greater than T_d.

3.2. Decoding the Watermark

Once the transformation matrix $\boldsymbol{A}$ has been detected the decoding of the watermark is straightforward and proceeds as follows.

1. Calculate the FFT of the windowed image $\boldsymbol{I}_w$ of size (I_m, I_n).
2. Generate the sequence of points $(x_1, y_1)...(x_{M_c}, y_{M_c})$ pseudo-randomly as determined by the secret key used during embedding.

3. Calculate the normalized coordinates in Fourier domain of the points as follows $(x_{ni}, y_{ni}) \rightarrow (x_i/1024, y_i/1024)$.

4. Apply the transformation matrix $\boldsymbol{A}$ to the normalized coordinates to yield $(\tilde{x}_1, \tilde{y}_1)...(\tilde{x}_{M_c}, \tilde{y}_{M_c})$

5. Extract the watermark from the transformed coordinates, taking into account the 90° coding scheme used during embedding and using bilinear interpolation to obtain values for samples which do not correspond directly to samples directly on the calculated FFT. This yields the bipolar signal $\tilde{\boldsymbol{m}}'$.

6. We then take the sign of $\tilde{\boldsymbol{m}}'$ and apply the transformation $-1 \rightarrow 0$ and $1 \rightarrow 1$ to yield the recovered binary bit sequence $\boldsymbol{b}$.

7. The bit sequence $\boldsymbol{b}$ represents the recovered message encoded by the BCH error correcting codes. This sequence is now decoded to yield the recovered message $\boldsymbol{m}_r$. For a message of length 72, if there are fewer than 5 errors, the 60 bit recovered message $\boldsymbol{m}_r$ will be identical to the embedded message $\boldsymbol{m}$ since these errors will be corrected by the BCH codes.

We note that in the first step we do not pad the image with zeros sine this leads to artifacts in the DFT domain. Rather, we perform directly the FFT on the windowed image and then work in normalized frequencies. We note that when performing the FFT, it is not necessary for the image size to be a power of 2 in order to transform from the spatial domain to DFT and vice-versa since we adopt the FFTW package[6] to calculate FFTs of arbitrary size efficiently.

4. RESULTS

In this section we evaluate the proposed approach relative to a standard series of tests detailed by Petitcolas and Kutter[15,9] and then compare the results to the performance of two commercially available algorithms. We first discuss the setting of the parameters of the algorithm and then proceed to present an extensive series of test results

4.1. Setting of Parameters

The algorithm contains several parameters which must be carefully set in order to obtain a robust watermark. During embedding the frequency band used is given by $f_1 = 717$ and $f_2 = 758$. In normalized

frequencies this corresponds to 0.35 and 0.37. This choice is made since it yields a good compromise between visibility and robustness. The images have been watermarked to yield a PSNR no greater than 38dB. This is done in order to yield a fair comparison between different methods. During detection we set $K_{min} = 0.5$ and $K_{max} = 2$. This is roughly equivalent restricting the range of allowable scale change in the image. We ignore scale changes smaller than $\frac{1}{K_{max}} = 0.5$ since the watermark is severely distorted and usually undecodable even if the template can be detected. Similarly large increases in size $\frac{1}{K_{max}} = 2$ are also ignored. The number of bins $N_b = 90$. The threshold for matching radial coordinates is set to $thresh = 0.002$, the minimum number of matches per line $N_m = 5$ and the detection threshold is set to $1 * 10^{-6}$. These last three parameters in practice yield excellent results where the detection of the template is concerned. In particular, when tested on several unmarked images, no false positives were encountered and examination of the benchmark results reveals that the transformations were in general well detected.

4.2. Test Results

We use the stirmark[14] program to evaluate the algorithm. The tests are divided into the following 8 sections: signal enhancement, compression, scaling, cropping, shearing, rotation, row/column removal, and random geometric distortions. We use the images of Lena, Mandrill and Fishingboat. The original and watermarked images appear in figures 3 and 4 respectively. We note that for a PSNR of 38dB, the watermark is invisible. For each attack we consider the attack by itself and where applicable after JPEG compression at a quality factor of 90. For each image we assign a score of 1 if for that case, the watermark is correctly decoded. If the watermark is incorrectly decoded, we assign a value of 0.

The results appear below in tables 1-5. We summarize the results in table 5 where we compute the average for each section. We note that for the compression section we first calculate the average for JPEG compression and then compute the average of the results with the results of GIF compression as done in Petitcolas' benchmark tests for the commercially available watermarking packages Digimarc[†] and Suresign[‡].

Relative to the benchmark series of tests, the watermark performs well and comparably to commercially available algorithms. The algorithm fails for random geometric distortions since the FFT is severely

[†]Digimarc Batch Embedding Tool c01.00.13 and Readmarc v1.5.8 used for the tests

[‡]SureSign Server version 1.94 used for the tests

distorted. However, to our knowledge, at this time no algorithm systematically decodes watermarks successfully after being attacked by the random geometric distortions implemented in the Stirmark3[14] package.

The major improvement lies in the fact that the algorithm recovers general affine transforms. We note in particular that the algorithm is successful 100% of the time in cases of shearing whereas the other algorithms only recover the watermark in cases where the shearing is small. Since the general affine transformation is not included in version 3.0 of the Stirmark benchmark tests used for our results, we include the examples in figure 5 which contain the watermarked image which has undergone a relatively large general linear transformation given by the matrix $\boldsymbol{A} = \begin{bmatrix} 1.3 & 0.1 \\ -0.05 & 0.8 \end{bmatrix}$. In all cases our algorithm successfully decodes the watermark.

5. CONCLUSION

In this article we have described a new algorithm for recovering watermarks which have undergone an arbitrary linear transformation. The main idea consists of adding structure to the template. This structure is exploited during decoding to yield a fast decoding algorithm. The method is robust against a wide variety of tests as indicated by the results obtained when evaluated relative to the extensive series of benchmark tests proposed in.[9]

ACKNOWLEDGMENTS

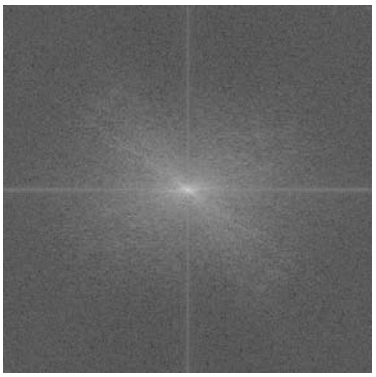
We thank Gabriela Csurka, Frederic Deguillaume, and Svyatoslav Voloshynovsky for there valuable insights. We also thank Fabien Petitcolas for making test results available for comparison and for providing the Stirmark3 software package. We are also grateful to Dr. Alexander Herrigel and Digital Copyright 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 Michael Frydrych, Jussi Parkkinen, and Ari Visa, editors, *The 10th Scandinavian Conference on Image Analysis*, pages 157–163, Lappeenranta, Finland, June 1997. Pattern Recognition Society of Finland.
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 16-19 1996.
3. S Craver, N Memon, BL Yeo, and MM Yeung. Can invisible watermark resolve rightful ownerships? In *Fifth Conference on Storage and Retrieval for Image and Video Database*, volume 3022, pages 310–321, San Jose, CA, USA, February 1997.
4. F. Deguillaume, G. Csurka, J. J. K. Ó Ruanaidh, and T. Pun. Robust 3d dft video watermarking. In *IS&T/SPIE Electronic Imaging'99, Session: Security and Watermarking of Multimedia Contents*, San Jose, CA, USA, January 1999.
5. J. F. Delaigle, C. De Vleeschouwer, and B. Macq. Watermarking algorithm based on a human visual model. *Signal Processing*, 66:319–335, 1998.
6. Matteo Frigo and Steven Johnson. fftw-1.3. MIT, Boston, Massachusetts, 1997-98.
7. 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.
8. C.-T. Hsu and J.-L. Wu. Hidden digital watermarks in images. *IEEE Transactions on Image Processing*, 8(1):58–68, January 1999.
9. M. Kutter and F. A. P. Petitcolas. A fair benchmark for image watermarking systems. In *Electronic Imaging '99, Security and Watermarking of Multimedia Contents*, volume 3657, pages 219–239, San Jose, CA, USA, January 1999.
10. 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.
11. J. J. K. Ó Ruanaidh and G. Csurka. A bayesian approach to spread spectrum watermark detection and secure copyright protection for digital image libraries. In *IEEE Conf. on Computer Vision and Pattern Recognition*, Fort Collins, Colorado, USA, June 1999.
12. 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.).

13. S. Pereira, J. J. K. Ó Ruanaidh, F. Deguillaume, G. Csurka, and T. Pun. Template based recovery of Fourier-based watermarks using Log-polar and Log-log maps. In *Int. Conference on Multimedia Computing and Systems, Special Session on Multimedia Data Security and Watermarking*, Juin 1999.
14. F. A. P. Petitcolas. http://www.cl.cam.ac.uk/ fapp2/watermarking/stirmark/. In *Stirmark3.0(60)*, 1999.
15. F. A. P. Petitcolas and R. J. Anderson. Attacks on copyright marking systems. In *2nd International Information Hiding Workshop*, pages 219–239, Portland, Oregon, USA, April 1998.
16. 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.
17. 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.
18. G. B. Rhoads. Steganography systems. In *International Patent WO 96/36163 PCT/US96/06618*, November 1996.
19. C. Britton Rorabaugh. *Error Coding Cookbook : Practical C/C++ Routines and Recipes for Error Detection and Correction*. McGraw Hill Text, 1996.
20. M. D. Swanson, B. Zhu, and A. H. Tewfik. Multiresolution scene-based video watermarking using perceptual models. *IEEE Journal on Selected Areas in Communications*, 16(4):540–550, May 1998.
21. A. Z. Tirkel, C.F. Osborne, and T.E. Hall. Image and watermark registration. *Signal processing*, 66:373–383, 1998.
22. George Voyatzis and Ioannis Pitas. Protecting digital image copyrights: A framework. *IEEE Computer Graphics and Applications*, 19(1):18–23, January 1999.

(a) LENA



(b) LOG OF FFT

Figure 1. ORIGINAL LENA IMAGE AND LOG OF MAGNITUDE OF FFT

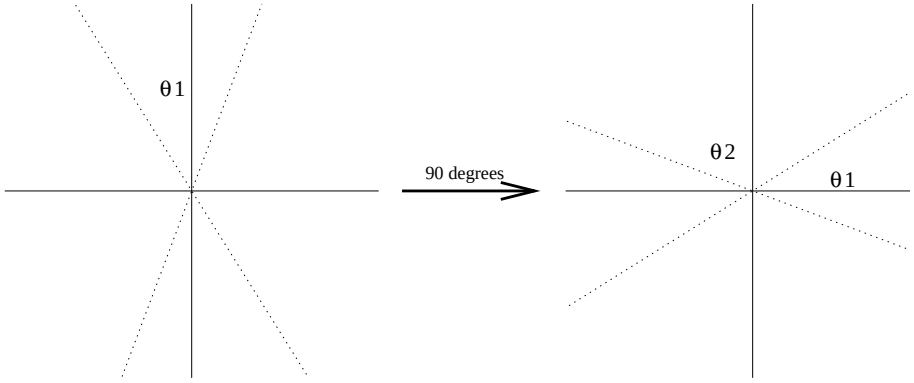
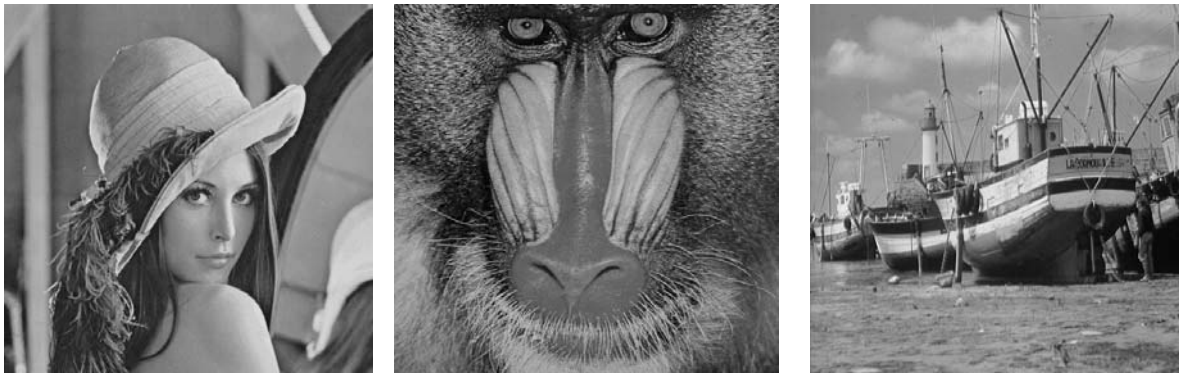
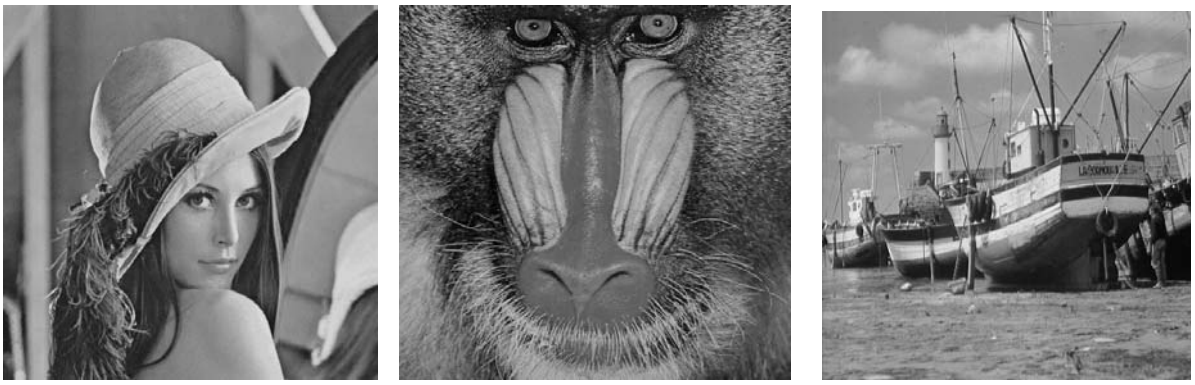


Figure 2. TEMPLATE AFTER 90° ROTATION



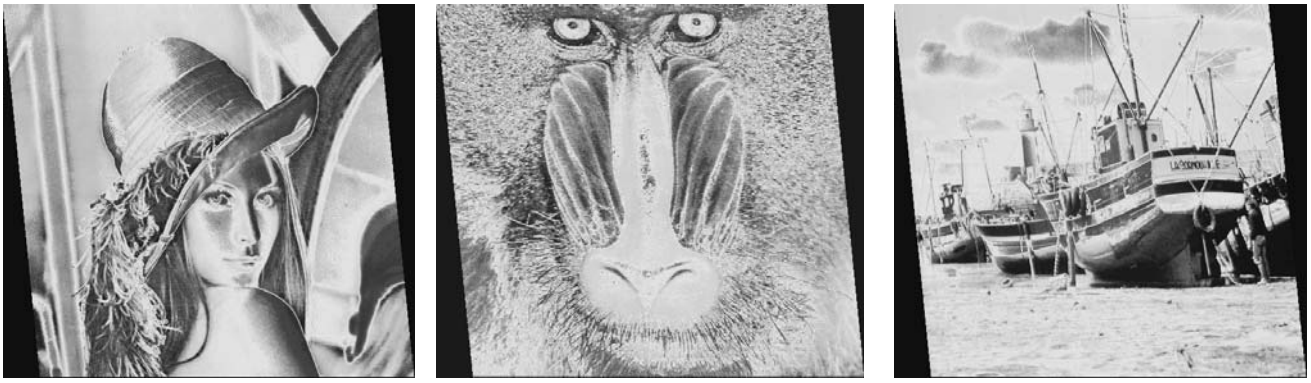
(a) LENA (b) MANDRILL (c) FISHINGBOAT

Figure 3. ORIGINAL IMAGES



(a) LENA (b) MANDRILL (c) FISHINGBOAT

Figure 4. WATERMARKED IMAGES WITH PSNR> 38dB



(a) LENA (b) MANDRILL (c) BOAT

Figure 5. WATERMARKED IMAGES WHICH HAVE UNDERGONE AN AFFINE TRANSFORMATION

	Lena	Mandrill	Fishingboat
Gaussian filter	1	1	1
Median filter	1	1	1
Sharpening	1	1	1
Frequency Mode Laplacian Removal	1	1	1

Table 1. RESULTS: SIGNAL ENHANCEMENT

JPEG	Lena	Mandrill	Fishingboat
10	0	0	0
15	0	0	0
25	0	0	0
50	0	0	0
60	0	0	0
75	1	1	1
80	1	1	1
85	1	1	1
90	1	1	1
GIF	1	1	not applicable

Table 2. RESULTS: COMPRESSION

Scale	Lena	Mandrill	Fishingboat
0.5	0	0	0
0.75	1	1	1
0.9	1	1	1
1.1	1	1	1
1.5	1	1	1
2	0	1	1
With JPEG			
0.5	0	0	0
0.75	1	1	1
0.9	1	1	1
1.1	1	1	1
1.5	1	1	1
2	0	1	1

Table 3. RESULTS: SCALING

% Cropped	Lena	Mandrill	Fishingboat
1	1	1	1
2	1	1	1
5	1	1	1
10	1	1	1
15	1	1	1
20	1	1	1
25	1	1	1
50	1	1	1
75	0	0	0
With JPEG			
1	1	1	1
2	1	1	1
5	1	1	1
10	1	1	1
15	1	1	1
20	1	1	1
25	1	1	1
50	1	1	1
75	0	0	0

Table 4. RESULTS: CROPPING

X-Shear	Lena	Mandrill	Fishingboat
$x \to 1.01x$	1	1	1
$x \to 1.1x$	1	1	1
With JPEG			
$x \to 1.01x$	1	1	1
$x \to 1.1x$	1	1	1
Y-Shear			
$y \to 1.01y$	1	1	1
$y \to 1.1y$	1	1	1
With JPEG			
$y \to 1.01y$	1	1	1
$y \to 1.1y$	1	1	1

Table 5. RESULTS: SHEARING

Rotation with Crop	Lena	Mandrill	Fishingboat
-2	1	1	1
-1	1	1	1
-0.5	1	1	1
0.5	1	1	1
1	1	1	1
2	1	1	1
5	1	1	1
10	1	1	1
15	1	1	1
30	1	1	1
45	1	1	1
90	1	1	1
With JPEG			
-2	1	1	1
-1	1	1	1
-0.5	1	1	1
0.5	1	1	1
1	1	1	1
2	1	1	1
5	1	1	1
10	1	1	1
15	1	1	1
30	1	1	1
45	1	1	1
90	1	1	1

Table 6. RESULTS: ROTATION WITH AUTO-CROP TO REMOVE ZERO-PADDED REGIONS

Rotation with Scale and Crop	Lena	Mandrill	Fishingboat
-2	1	1	1
-1	1	1	1
-0.5	1	1	1
0.5	1	1	1
1	1	1	1
2	1	1	1
5	1	1	1
10	1	1	1
15	1	1	1
30	1	1	1
45	1	1	1
90	1	1	1
With JPEG			
-2	1	1	1
-1	1	1	1
-0.5	1	1	1
0.5	1	1	1
1	1	1	1
2	1	1	1
5	1	1	1
10	1	1	1
15	1	1	1
30	1	1	1
45	1	1	1
90	1	1	1

Table 7. RESULTS: ROTATION WITH SCALE TO ORIGINAL SIZE THEN CROP TO REMOVE ZERO-PADDED REGIONS

Row/Column Removal	Lena	Mandrill	Fishingboat
1	1	1	1
5	1	1	1
10	1	1	1
With JPEG			
1	1	1	1
5	1	1	1
10	1	1	1
Flip	1	1	1

Table 8. RESULTS: ROW/COLUMN REMOVAL AND FLIP

Random Geometric Distortions	Lena	Mandrill	Fishingboat
	0	0	0

Table 9. RESULTS: RANDOM GEOMETRICAL DISTORTIONS IMPLEMENTED BY STIRMARK[14]

	Proposed approach	Digimarc	Suresign
Enhancement	1	1	1
Compression	0.74	0.81	0.95
Scaling	0.78	0.72	0.95
Cropping	0.89	1	1
Shearing	1	0.5	0.5
Rotation	1	0.94	0.5
Row/column removal+flip	1	1	1
Random Geometrical Distortions	0	0.33	0

Table 10. RESULTS: SUMMARY AND COMPARISON

FIGURES:

1. ORIGINAL LENA IMAGE AND LOG OF MAGNITUDE OF FFT
2. TEMPLATE AFTER 90° ROTATION
3. ORIGINAL IMAGES
4. WATERMARKED IMAGES WITH PSNR> $38dB$
5. WATERMARKED IMAGES WHICH HAVE UNDERGONE AN AFFINE TRANSFORMATION

TABLES:

1. RESULTS: SIGNAL ENHANCEMENT
2. RESULTS: COMPRESSION
3. RESULTS: SCALING
4. RESULTS: CROPPING
5. RESULTS: SHEARING
6. RESULTS: ROTATION WITH AUTO-CROP TO REMOVE ZERO-PADDED REGIONS
7. RESULTS: ROTATION WITH SCALE TO ORIGINAL SIZE THEN CROP TO REMOVE ZERO-PADDED REGIONS
8. RESULTS: ROW/COLUMN REMOVAL AND FLIP
9. RESULTS: RANDOM GEOMETRICAL DISTORTIONS IMPLEMENTED BY STIRMARK[14]
10. RESULTS: SUMMARY AND COMPARISON