Fast Robust Template Matching for Affine Resistant Image Watermarks

Shelby Pereira and Thierry Pun[1]

University of Geneva - CUI,
24 rue General Dufour, CH 1211
Geneva 4, Switzerland,
{Shelby.Pereira,Thierry.Pun}@cui.unige.ch
WWW home page: http://cuiwww.unige.ch/ vision

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 for the accurate and efficient recovery of the template in an image which has undergone a general affine transformation. Furthermore we demonstrate how the template can be used as a tool for asserting the presence of a watermark. We also systematically evaluate the algorithm and 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. 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 log-polar or log-log-maps, we propose searching the space of possible affine transformations. Since an exhaustive search leads to an intractable problem, we demonstrate how a careful pruning of the search space yiels to robust detection of transformations reasonable quickly. 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

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

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

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

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:

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

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

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[1] 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.

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.

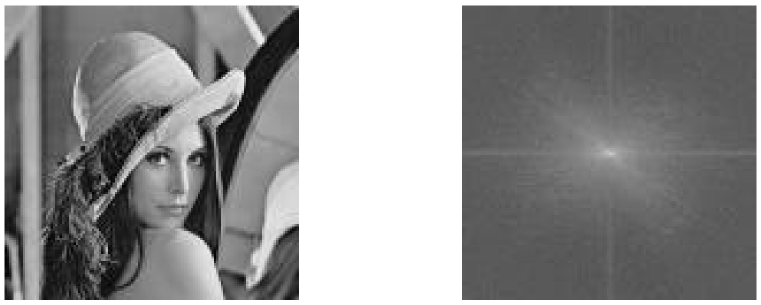
$$F(k_1, k_2) \exp\left[-j(ak_1 + bk_2)\right] \leftrightarrow f(x_1 + a, x_2 + b) \tag{8}$$

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 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.

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



(a) LENA (b) LOG OF FFT

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

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]. We have found experimentally that using templates of approximately 8 points works best. In section 3.1 we provide a theoretical justification using probabilistic arguments to justify this choice. The points of the template are distributed uniformly in the DFT 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 three 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. We also note that it is critical that the points be inserted strongly enough so that they remain peaks after interpolation errors from possible transformations. The reason for this will be clear in the next section.

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. The main idea is to search possible transformations while taking care not to examine the large amount of cases that cannot lead to a successful template match.

3.1 Template Detection

The template detection process involves several steps which are enumerated below. We first present the basic algorithm and then indicate how to effectively prune the search space by applying a few heuristics.

1. Calculate the FFT of the image zero padded to 1024×1024.
2. Extract the positions of all the local peaks (p_{xi}, p_{yi}) in the image. We denote this set of peaks as $\boldsymbol{P}$.
3. Choose two points in the template (x'_1,y'_1) and (x'_2,y'_2).
4. For all pairs of points $(p_{xi}, p_{yi}); (p_{xj}, p_{yj})$ perform the following steps
   (a) Compute the transformation matrix $\boldsymbol{A}$ which maps the two template points to the point pair $(p_{xi}, p_{yi}); (p_{xj}, p_{yj})$
   (b) Apply the transformation to the other template points
   (c) Count the number of transformed template points lie within a small radius r_{min} of any peak in the set $\boldsymbol{P}$. These are the matched template points
   (d) If we have at least N_m matches we conclude that the template is found and terminate the search, otherwise we proceed to the next point pair.
5. If all point pairs have been tested and we never obtain N_m matches, we conclude that no watermark has been found.
6. If we have at least N_m matches, recalculate the linear transformation $\boldsymbol{A}$ using all the matched points such that the mean square estimation error in equation 9 is minimized.

$$mse = \frac{1}{nummatches} \| \boldsymbol{A} \begin{bmatrix} x'_1 & y'_1 \\ \vdots & \vdots \\ x'_l & y'_l \end{bmatrix}^T - \begin{bmatrix} p_{x1} & p_{y1} \\ \vdots & \vdots \\ p_{xl} & p_{yl} \end{bmatrix}^T \|^2 \tag{9}$$

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 $\boldsymbol{A}$. If we have less than N_m matches we conclude that no watermark is found.

Some observations are necessary. In step 1 we pad to 1024×1024 in order to obtain a good resolution in the FFT domain. This is imporant in order to obtain accurate estimates of the transformation matrix. We also note that the re-estimation in step 6 is important since it increases the accuracy of the matrix $\boldsymbol{A}$. Finally we note that step 4d 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.

Step 4 contains the most costly part of the algorithm. As it stands the cost of the algorithm is $O(N^2 M)$ where N is the number of points in $\boldsymbol{P}$ and M is the number of points in the template. N can be as large as 500 points so that we find that the algorithm may take up to 5 minutes in some cases on a pentium 400MHz machine to find a search.

However the search space can be drastically pruned by exploiting two heuristics. Firstly we observe that if we choose in step 5 the points (x_1',y_1') and (x_2',y_2') which are (r_1',θ_1') and (r_2',θ_2') in polar coordinates then if $r_1' > r_2'$ we need only consider points in the $\boldsymbol{P}$ where $r_1 > r_2$. Similarly if $r_1' < r_2'$ we need only consider points in the $\boldsymbol{P}$ where $r_1 < r_2$. Here we are in fact exploiting the fact that for realistic transformations small and large frequencies will never be inverted. This immediately reduces the search space by a factor of 2.

The second important observation is that for transformations which are likely to occur in practice the difference in θ between two template points will change only slightly. In practice we can limit the change in θ to roughly $\pm 20°$. Exploiting this observation further reduces the search space to $\frac{40°}{360°}$ of the original size. When we apply these heuristics we obtain an algorithm which finds the affine transformation in roughly 15 seconds on a pentium 400MHz.

Since we are using the template to assert the presence of a watermark, it is important to evaluate the probability of a false detection in order to justify the appoach. The evaluation of this probability is relatively straightforward. We first note that on a 1024×1024 grid with 500 points (which for simplicity we assume are uniformly distributed) the probability of finding a point in a given location is $500/1024^2 \approx \frac{1}{2000}$. Since we work on a discrete grid, when we look for a match we also look in the 8 surrounding neighbours (i.e. r_{min} equals one pixel in step 5c), we multiply by 9 to obtain $\frac{9}{2000}$. We must now count the number of transformtaions which will be calculated. If we ignore the heuristics used to prune the search space, an exhaustive search involves $2N^2 = 2 \times 500^2$ transformations.

The factor of 2 comes from the fact that the ordering is essential. By pruning the search space, we reduce the number of calculated transformations by a factor of 2×9 so that roughly 3000 transformations are evaluated. Now if we embed a template with 8 points and insist that all 8 points be matched at detection, we obtain that the probability of a false match given by equation 10. This probablity is extremely small and in practice no false detections have been encountered.

$$P_{false} = \frac{9}{2000}^{8} \times 3000 \approx 5.0 \times 10^{-16} \tag{10}$$

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
$(x_{ni}, y_{ni}) = (x_i/1024, y_i/1024)$.
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 $-1 \to 0$ and $1 \to 1$ 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.

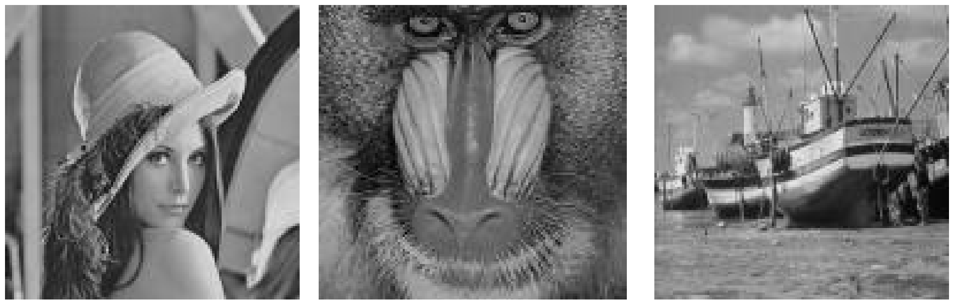
We note that in the first step we do not pad the image with zeros since this leads to artifacts in the DFT domain. Rather, we perform directly the FFT on the windowed image and then work in normalized frequencies. We note that when performing the FFT, it is not necessary for the image size to be a power 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.

4.1 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 2 and 3 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. We then compute an average for each section and summarize the results.

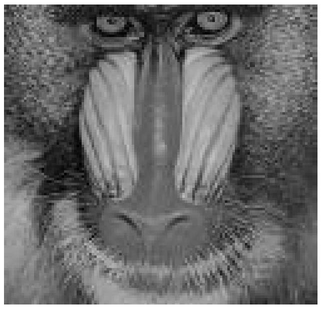


(a) LENA (b) MANDRILL (c) FISHINGBOAT

Fig. 2. ORIGINAL IMAGES

In the signal enhancement section which includes Gaussian, median, and sharpening filters as well as the Frequency Mode Laplacian Removal attack, the watermark was correctly decoded in all cases. The algorithm is successful against JPEG down to a level of 75 quality factor. In all cases the combinations of rotations, scales and cropping were correctly recovered. Against scaling the watermark is successlfully recovered in the range 0.75 to 2, but fails when the image is scaled to 0.5. This can be expected since 75% of the information is lost in this case. The watermarked is recovered when up to 50% of the image has been cropped. Furthermore, the watermark is also successfully recovered against shearings of 0.01% and 0.1% and combinations of row and column removal. Unfortunately the watermark is never recovered when the random geometric distortions implemented by the stirmark program are applied. We summarize the results in table 4.1 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

(a) LENA



(b) MANDRILL

(c) FISHINGBOAT

Fig. 3. WATERMARKED IMAGES WITH PSNR> 38dB

	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 1. RESULTS: SUMMARY AND COMPARISON

compression as done in Petitcolas' benchmark tests for the commercially available watermarking packages Digimarc[2] and Suresign[3].

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 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 tested the algorithm on images which have undergone a relatively

[2] Digimarc Batch Embedding Tool c01.00.13 and Readmarc v1.5.8 used for the tests
[3] SureSign Server version 1.94 used for the tests

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 pruning a relatively large search space to obtain a fast decoding algorithm which recoveres the general affine transformation. 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] and is comparable to currently available commercial algorithms with the major improvement lies in its ability to detect general affine transformations whereas other algorithms are generally limited to rotations and scales.

ACKNOWLEDGMENTS

We thank Gabriela Csurka, Frederic Deguillaume, and Svyatoslav Voloshynovski 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.