Rotation, Scale, and Translation Resilient Watermarking for Images

Ching-Yung Lin, *Member, IEEE*, Min Wu, *Student Member, IEEE*, Jeffrey A. Bloom, *Member, IEEE*, Ingemar J. Cox, *Senior Member, IEEE*, Matt L. Miller, and Yui Man Lui, *Member, IEEE*

***Abstract*—Many electronic watermarks for still images and video content are sensitive to geometric distortions. For example, simple rotation, scaling, and/or translation (RST) of an image can prevent blind detection of a public watermark. In this paper, we propose a watermarking algorithm that is robust to RST distortions. The watermark is embedded into a one-dimensional (1-D) signal obtained by taking the Fourier transform of the image, resampling the Fourier magnitudes into log-polar coordinates, and then summing a function of those magnitudes along the log-radius axis. Rotation of the image results in a cyclical shift of the extracted signal. Scaling of the image results in amplification of the extracted signal. And translation of the image has no effect on the extracted signal. We can therefore compensate for rotation with a simple search, and compensate for scaling by using the correlation coefficient as the detection measure.**

False positive results on a database of 10 000 images are reported. Robustness results on a database of 2000 images are described. It is shown that the watermark is robust to rotation, scale, and translation. In addition, we describe tests examining the watermarks resistance to cropping and JPEG compression.

***Index Terms*—Fourier–Mellin, rotation, RST, scale, translation, watermarking.**

I. Introduction

THERE has been much emphasis on the robustness of watermarks to common signal processing operations such as compression and signal filtering. However, recently it has become clear that even very small geometric distortions can prevent the detection of a watermark [1], [2]. This problem is most pronounced when the original unwatermarked image is unavailable to the detector. Conversely, if the original image is available to the detector, then the watermarked image can often be registered to the original and the geometric distortion thereby inverted.[1] Blind detection requires that detection of the watermark be performed without access to the original unwatermarked image. As such, it is not possible to invert the geometric distortion based on registration of the watermarked and original images.

Before proceeding further, it is important to define what we mean by the geometric distortions of rotation, scale, and translation. Specifically, we are interested in the situation in which a watermarked image undergoes an *unknown* rotation, scale, and/or translation prior to the detection of the watermark. The detector should detect the watermark if it is present. This definition is somewhat obvious, so it may be more useful to describe what we are not interested in. In particular, some watermark algorithms claim robustness to scale changes by first embedding a watermark at a canonical scale, then changing the size of the image, and finally, at the detector, scaling the image back to the canonical size prior to correlation. In our opinion, that detector does not see a scale change. Rather, the process is more closely approximated by a low pass filtering operation that occurs when the image is reduced in size. In the scaling degradation with which we are concerned, the detector is unaware of the scaling and cannot rescale or pad to the original size. Similarly, tests that rotate an image by some number of degrees and subsequently rotate the image by the same amount in the opposite direction are not adequate tests of robustness to rotation. The same is true for translation. The common situation we are concerned with occurs when a watermarked image is printed and then cropped or padded and scanned back into the digital domain. In these circumstances, the image dimensions have changed both because of cropping and possibly scaling. There is also likely to be an associated translational shift. We assume that the detector is not informed of the rotation, scale, and translation parameters. In this example, scaling to a canonical size does not undo the scaling. Rather, if the cropping is not symmetric in both the rows and columns, then scaling to a canonical size will result in a change in the image's aspect ratio. Changes in aspect ratio are not addressed in this paper. Application of the current watermarking method to the print and scan process is discussed elsewhere [5].

One strategy for detecting watermarks after geometric distortion is to try to identify what the distortions were, and invert them before applying the watermark detector. This can be accomplished by embedding a registration pattern along with the watermark [6], [7].

One problem with this solution is that, because it requires the insertion of a registration watermark in addition to the data-carrying watermark, this approach is likely to reduce the image fidelity. A second problem arises because all images watermarked with this method will share a common registration watermark. This fact may improve collusion attempts to discern the regis-

Manuscript received February 29, 2000; revised December 5, 2000. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Naohisa Ohta.

C.-Y. Lin is with IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA.

M. Wu is with Princeton University, Princeton, NJ 08543 USA.

J. A. Bloom is with the Sarnoff Corporation, Princeton, NJ 08543 USA (e-mail: bloom@research.nj.nec.com).

I. J. Cox and M. L. Miller are with the NEC Research Institute, Princeton, NJ 08540 USA.

Y. M. Lui is with Robotic Vision Systems, Inc., Happauge, NY 11749 USA.

Publisher Item Identifier S 1057-7149(01)02478-2.

[1]Although the original Cox *et al.* [3] algorithm did not include this step, subsequent commercial implementations did so. More recently, Johnson *et al.* [4] observed that it is not necessary to retain the entire image, a sufficiently small set of key points will suffice.

tration pattern and, once found, the registration pattern could be removed from all watermarked images thus restricting the invertibility of any geometric distortions.

Another way to implement the above strategy is to give the watermark a recognizable structure. For example, as suggested in [8]–[10], the watermark might be embedded multiple times in the image at different spatial locations. The autocorrelation function of a watermarked image will then yield a pattern of peaks corresponding to the embedded locations. Changes in this pattern of peaks can be used to describe any affine distortions to which the watermarked image has been subjected. This method has significant potential, but, similar to the above methods, has two failure modes. For successful detection both the identification of the geometric distortion and the detection of the watermark after inversion of that distortion must be successful. Both of these processes must be robust and resistant to tampering.

Yet another approach to address geometric distortions is the "normalization" of the image prior to watermark embedding. After embedding, the image is restored to its original geometric state prior to distribution. Upon receipt, the image is again normalized prior to detection. Unlike scaling to a canonical size, the normalization must be invariant to the expected geometric distortions. For example, in [11], images are normalized by their geometric moments.

Our proposal employs an alternative strategy based on developing a watermark that is invariant to geometric distortions, thus eliminating the need to identify and invert them. In particular, we are concerned with distortions due to rotation, scale, and/or translation (RST). While these geometric distortions have recently become of interest to the watermarking community, they have long been of interest to the pattern recognition community. A comprehensive discussion of the pattern recognition literature is outside the scope of this paper. Hu [12] described the use of moment invariants for visual pattern recognition of planar geometric figures. It has been shown [13] that these classic moment invariants are equivalent to the radial moments of circular-harmonic functions (CHF's) that arise from a Mellin transform of the log-polar representation of an image when the complex Mellin radial frequency s, is a real integer $s \geq 1$.

The Fourier–Mellin transform is closely related to the algorithm described in this paper. There are a variety of related ideas from pattern recognition. First, Casasent and Psaltis [14], [15] note that the signal-to-noise ratio of the correlation peak between two images decreases from 30 dB to 3 dB with either a 2% scale change or a 3.5° rotation. Their proposal is essentially a hybrid opto-electronic implementation of the Fourier–Mellin transform. Altmann and Reitbock [16] and Altmann [17] discuss implementation issues related to the discrete Fourier–Mellin transform. These include interpolation, aliasing, and spectral border effects, which are discussed in detail in Section III of this paper. Wechsler and Zimmerman [18] describe a conformal-log mapping that is very closely related to the Fourier–Mellin transform. Also, Lin and Brandt [19] discuss the use of the Fourier–Mellin and other transforms for pattern recognition. They describe a number of *absolute* or *strong* invariants based on the phase of the Fourier or Fourier–Mellin spectrums. The terms "absolute" and "strong" refer to the fact that all information about an image except that of position, orientation, or scale is preserved. This may be important for recognition tasks, especially if the library of objects is large. Ferraro and Caelli [20] discuss this issue in more detail.

While strong invariance might be required in an object recognition application, we do not believe that this property is necessary for watermarking applications. From a strongly invariant representation, an image can be fully reconstructed modulo rotation, scale, and translation. However, distinguishing between watermarks can be accomplished in a much lower dimensional space than can distinguishing between images. Therefore we can consider invariant transformations that may be lossy as the image is projected into a lower dimensional space. The reasons we can distinguish between watermarks in fewer dimensions are two-fold. First, the set of watermarks for which a detector might search is relatively small compared to a typical object recognition database. Second, a watermark is not a naturally occurring object but is artificially inserted into an image. As such, the watermark can be designed to be easily represented. In particular, it is often advantageous to represent the watermark as a one-dimensional (1-D) projection of the image space. If properly designed, this has the benefit of reducing a two-dimensional (2-D) search to one dimension, thereby significantly reducing the computational cost.

O'Ruanaidh and Pun [21] first suggested a watermarking method based on the Fourier–Mellin transform. However, they note very severe implementation difficulties which we suspect have hampered further work in this area. They choose to use a transformation that is strongly invariant claiming that "it is more convenient to use strong invariants because the last stage of embedding a mark involves inverting the invariant representation to obtain the watermarked image." We believe that invertibility is not essential. Following the formulation in [22], suppose we have a noninvertible extraction function, $X(C)$, that maps a Work, C, into an extracted signal. Such a function would be used as part of a detection strategy. An example extraction function found in the literature [23] is

$$X_i(C) = \sum_{j \in R_i} C(j) \qquad 1 \leq i \leq N \tag{1}$$

where R_i are disjoint subsets of elements of the Work, C. We can often define an embedding function, $Y(w, C)$, which finds a new Work, $C_w = Y(w, C_o)$, such that

$$X(C_w) \equiv X(Y(w, C_o)) \approx w \tag{2}$$

and C_w is perceptually similar to C_o. In other words, the watermarked image looks like the original and the vector extracted during detection *looks like* the watermark vector. This function is sufficient for use in a watermark embedder.

There have been a number of other recent watermarking algorithms designed to deal with geometric distortions. Of particular note is the recent work of Bas *et al.* [24]. They describe an algorithm based on the detection of salient features in an image and the insertion of signals relative to these salient features. Experimental results indicate that the method is robust to mirror reflection and rotation. However, surprisingly, the system fails to survive other geometric distortions. A somewhat related set

of methods is described by Maes and van Overveld [25] and Rongen *et al.* [26]. These methods are based on geometrically warping local regions of an image onto a set of random lines. However, currently, these methods are not robust to geometric distortions, but rather, allow for a rapid, but exhaustive search through the possible set of geometric distortions.

In Section II we describe our algorithm. It differs from that of [21] in two important ways. First, we choose to watermark a projection of the transform space. Second, the watermark embedding is based on the principle of communication with side information [22]. This is described in more detail in Section III, including the iterative procedure for dealing with the one-to-many mapping from watermark space to image space. Our solutions to a number of implementation issues are also discussed in Section III. Section IV describes the results of experiments on a large database.

II. Algorithm

Consider an image $i(x, y)$ and a rotated, scaled, and translated version of this image, $i'(x, y)$. Then we can write

$$i'(x,y) = i(\sigma(x\cos\alpha + y\sin\alpha) - x_0, \sigma(-x\sin\alpha + y\cos\alpha) - y_0) \quad (3)$$

where the RST parameters are α, σ, and (x_0, y_0) respectively.

The Fourier transform of $i'(x, y)$ is $I'(f_x, f_y)$, the magnitude of which is given by

$$|I'(f_x,f_y)| = |\sigma|^{-2}|I(\sigma^{-1}(f_x\cos\alpha + f_y\sin\alpha), \sigma^{-1}(-f_x\sin\alpha + f_y\cos\alpha))|. \quad (4)$$

Equation (4) is independent of the translational parameters, (x_0, y_0). This is the well known translation property of the Fourier transform [27].

If we now rewrite (4) using log-polar coordinates, i.e.,

$$f_x = e^\rho\cos\theta \quad (5)$$

$$f_y = e^\rho\sin\theta \quad (6)$$

then the magnitude of the Fourier spectrum can be written as

$$|I'(f_x,f_y)| = |\sigma|^{-2}|I(\sigma^{-1}e^\rho\cos(\theta-\alpha), \sigma^{-1}e^\rho\sin(\theta-\alpha))| \quad (7)$$

$$= |\sigma|^{-2}|I(e^{(\rho-\log\sigma)}\cos(\theta-\alpha), e^{(\rho-\log\sigma)}\sin(\theta-\alpha))| \quad (8)$$

or

$$|I'(\rho,\theta)| = |\sigma|^{-2}|I(\rho-\log\sigma, \theta-\alpha)|. \quad (9)$$

Equation (9) demonstrates that the amplitude of the log-polar spectrum is scaled by $|\sigma|^{-2}$, that image scaling results in a translational shift of $\log\sigma$ along the ρ axis, and that image rotation results in a cyclical shift of α along the θ axis.

We need not be concerned with the amplitude scaling of the spectrum, since we intend to perform watermark detection using the correlation coefficient, which is invariant to this scaling. See Section II-A for more details.

Next, we define $g(\theta)$ to be a 1-D projection of $|I(\rho, \theta)|$ such that

$$g(\theta) = \sum_j \log(|I(\rho_j, \theta)|). \quad (10)$$

The reason for summation of the log values rather than the magnitudes themselves is discussed in Section III-D. Due to the symmetry of the spectra of real images

$$|F(x,y)| = |F(-x,-y)| \quad (11)$$

we only compute $g(\theta)$ for $\theta \in [0^\circ \cdots 180^\circ)$.

We find it convenient to add the two halves of $g(\theta)$ together, obtaining

$$g_1(\theta') = g(\theta') + g(\theta' + 90^\circ) \quad (12)$$

with $\theta' \in [0^\circ \cdots 90^\circ)$. The reasons for this are discussed in Section III-F.

Clearly, $g_1(\theta)$, is invariant to both translation and scaling. However, rotations result in a (circular) shift of the values of $g_1(\theta)$. If θ is quantized to the nearest degree, then there are only 90 discrete shifts, and we handle this by an exhaustive search.

A. Watermark Detection Process

In principle, detectors may be built that can handle watermarks encoding several bits [41]. However, the present detector determines only whether or not a given watermark has been embedded in a given image. It takes as input, an image and a watermark and the output is a single bit indicating whether the image contains the watermark.

The watermark is expressed as a vector of length N. To determine whether the watermark is present, an "extracted signal" $v = g_1(\theta)$ is computed from the image, for N values of θ evenly spaced between 0° and 90°. The extracted signal is then compared to the watermark using the correlation coefficient. If the correlation coefficient is above a detection threshold T, then the image is judged to contain the watermark.[2].

Thus, the basic algorithm for watermark detection proceeds as follows.

1) Compute a discrete log-polar Fourier transform of the input image as described in Section III-A. This can be thought of as an array of K rows by N columns, in which each row corresponds to a value of ρ, and each column corresponds to a value of θ.
2) Sum the logs of all the values in each column, and add the result of summing column j to the result of summing column $j + N/2$ ($j = 0 \cdots ((n/2) - 1)$) to obtain an invariant descriptor v, in which

$$v_j = g_1(\theta_j) \quad (13)$$

where θ_j is the angle that corresponds to column j in the discrete log-polar Fourier transform matrix.

[2]The use of correlation coefficient as a detection measure is recommended in [22] One benefit of this metric is its independence to scaling of the signal amplitudes.

3) Compute the correlation coefficient D, between v and the input watermark vector w, as

$$D = \frac{w \cdot v}{\sqrt{(w \cdot w)(v \cdot v)}} \tag{14}$$

4) If D is greater than a threshold T, then indicate that the watermark is present. Otherwise, indicate that it is absent.

B. Watermark Embedding Process

Once a method for detecting watermarks has been defined, we can construct a watermark embedding algorithm according to the methodology described in [22]. In that paper, watermarking is cast as a case of communications with side information at the transmitter, which is a configuration studied by Shannon [28]. The difference between this view of watermarking, and a more common view, is as follows.

Most public watermarking methods found in the literature use blind embedding in that the original image is considered to be noise. The embedder adds a small-amplitude signal to this noise, and the detector must be sensitive enough to work with the small signal-to-noise ratio that results.

However, this common approach ignores the fact that the embedder has complete knowledge of the "noise" caused by the original image. If we view the embedder as a transmitter and the cover image as a communications channel, then this knowledge amounts to side-information about the behavior of that channel. When the transmitter knows ahead of time what noise will be added to the signal, its optimal strategy is to subtract that noise from the signal before transmission. The noise then gets added back by the communications channel, and the receiver receives a perfect reconstruction of the intended signal.

In the case of watermarking, it is unacceptable for the embedder to subtract the original image from the watermark before embedding the watermark, because it would result in unacceptable fidelity loss. In fact, if the watermark is expressed as a pattern that is the same size as the image, then this strategy simply replaces the image with the watermark pattern, which is clearly too drastic. However, when the watermark is expressed as a signal in a lower-dimensional space, as is the case with the present system, the results need not be so drastic, since a wide variety of full-resolution images project into the same extracted signal and the embedder may choose the one that most resembles the original. But even in the case of lower-dimensional watermarks, it is not always possible to completely replace the extracted signal with the watermark signal while maintaining acceptable fidelity.

To make maximal use of the side-information at the embedder, while maintaining acceptable fidelity, [22] introduces the idea of a "mixing function," $f(v, w)$. This takes an extracted signal v, and a watermark vector w, as input, and the output is a signal s, which is perceptually similar to v, and has a high correlation with w. Since s is something between v and w, it is referred to as the "mixed signal." It is this mixed signal that the embedder transmits, by modifying the image so that the extraction process in the detector will produce s.

The basic approach for embedding described in [22] consists of three steps:

1) Apply the same signal-extraction process to the unwatermarked image as will be applied by the detector, thus obtaining an extracted vector, v. In our case, this means computing $g_1(\theta)$.
2) Use the mixing function, $s = f(v, w)$, to obtain a mixture between v and the desired watermark vector, w. At present, our mixing function simply computes a weighted average of w and v, which is a highly sub-optimal approach. More sophisticated mixing methods, for example those examined in [29], may be used.
3) Modify the original image so that, when the signal-extraction process is applied to it, the result will be s instead of v. This process is implemented as follows:
   a) Modify all the values in column j of the log-polar Fourier transform so that their logs sum to s_j instead of v_j. This could be done, for example, by adding $(s_j - v_j)/K$ to each of the K values in column j. Care must be taken to preserve the symmetry of DFT coefficients.
   b) Invert the log-polar resampling of the Fourier magnitudes, thus obtaining a modified, Cartesian Fourier magnitude.
   c) The complex terms of the original Fourier transform are scaled to have the new magnitudes found in the modified Fourier transform.
   d) The inverse Fourier transform is applied to obtain the watermarked image.

Unfortunately, there is inherent instability in inverting the log-polar resampling of the Fourier magnitude (Step 3b). We therefore approximate this step with an iterative method in which a local inversion of the interpolation function is used for the resampling. The method is described in Section III-B.

III. Implementation Problems and Solutions

There are a number of problems that arise when implementing the algorithm of Section II. Several of these are addressed below.

A. Rectilinear Tiling Implied by DFT

The log-polar Fourier transform of an image can be computed by resampling the image DFT with a log-polar grid. Some interpolation method must be used during the resampling, since the log-polar sample points do not generally coincide with the Cartesian sample points in the DFT.

The DFT is conventionally assumed to represent a tiled version of an image, as illustrated in Fig. 1(a). Stone *et al.* [30] have noted that this tiling pattern represents an inherent problem for any algorithm that relies on the rotational properties of Fourier transforms, since, when the content of an image is rotated, the rectilinear tiling grid is not rotated along with it. Thus, the DFT of a rotated image is not the rotated DFT of that image. The problem is illustrated in Fig. 1(b) and (c).

One possible solution is to compute the log-polar Fourier transform directly, without using the Cartesian DFT as an intermediate step. In the continuous Fourier domain, each point has a value determined by correlating the image with a complex, planar sinusoid. If we wish to obtain a value for a point between

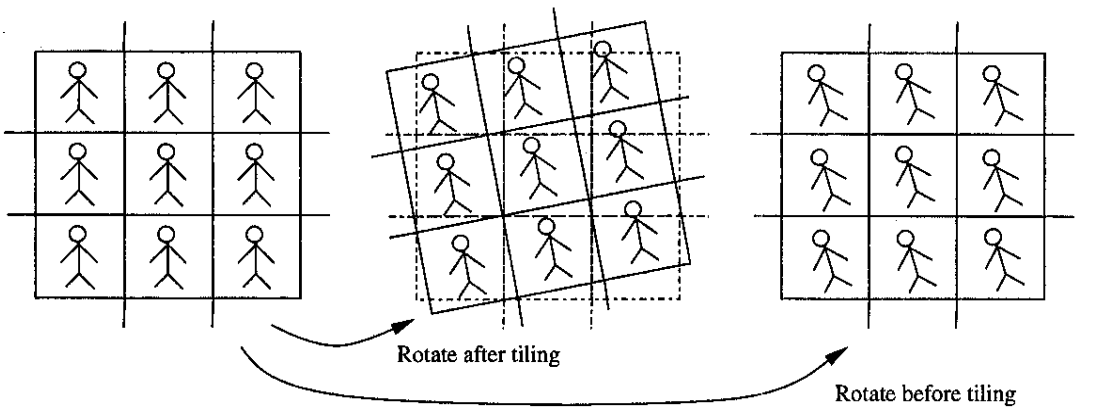


Fig. 1. Rectilinear tiling and image rotation.

those that would be sampled in a DFT, we can find the corresponding sinusoid and directly compute its correlation with the image. This amounts to assuming that all the pixel values outside the bounds of the image are black, rather than assuming they are tiled copies of the image. The same result can be obtained by applying a sinc interpolation to the Cartesian DFT coefficients in order to determine the values on the log-polar grid.

The direct approach described above does not take advantage of efficient methods available for computing DFTs. Both this and the sinc interpolation approach are thus likely to be prohibitively expensive.[3] Instead, we approximate the log-polar Fourier transform by using an inexpensive interpolation preceded by zero-padding as follows.

1) Pad the image with black to obtain a larger image.
2) Take the DFT of the padded image. This yields a more finely sampled version of the continuous Fourier transform.
3) Resample in a log-polar grid, using an inexpensive interpolation technique. The technique we use is linear interpolation of the magnitudes of the coefficients.

The zero-padding has the effect of adding separation between the implicit tiles in the spatial domain, thus reducing the distortions shown in Fig. 1. Viewed another way, by padding with black we obtain a denser sampling of the Fourier transform, thus reducing the distances between the DFT's sample points and the log-polar sample points and reducing the error introduced by inexpensive interpolation.

B. Difficulty of Inverting Log-Polar Mapping

Each element of the log-polar Fourier magnitude array is a weighted average of up to four elements of the Cartesian Fourier magnitude array. Thus, we can write

$$F = MC \tag{15}$$

where

F column vector containing all the elements of the log-polar array;

C column vector containing the elements of the Cartesian array;

M contains the weights used to perform interpolation.

If we wish to modify the log-polar array so that it contains the watermark, and then find the corresponding Cartesian array, we have to find the inverse of M. Unfortunately, M is ill-conditioned and it is not practical to perform this inversion precisely.

Instead, we use an iterative process to perform an approximate inversion. Let F' be the modified version of F (the results of Step 3a in Section II-B). We begin by observing that the four nonzero values in each row of M sum to 1. Thus, if we add $F_i' - F_i$ to each of the elements $C_{j_1} \cdots C_{j_4}$, where $M_{i,j_1} \cdots M_{i,j_4}$ are nonzero, then the resulting Cartesian array will yield F_i' in its log-polar mapping.

Unfortunately, if we try to apply this method to change all the elements of F, we'll have conflicting changes in the various elements of C. For example, both $M_{i,j}$ and $M_{k,j}$ might be nonzero, so that we'd need to change C_j both when changing F_i to F_i' and when changing F_k to F_k'. The desired changes are unlikely to be the same. We resolve this problem by using a weighted average of all the desired changes to each element of C. So, in the above example, we would change the value of C_j by

$$\frac{M_{i,j}(F_i' - F_i) + M_{k,j}(F_k' - F_k)}{M_{i,j} + M_{k,j}} \tag{16}$$

(assuming that $M_{i,j}$ and $M_{k,j}$ are the only nonzero elements of column j).

The above method results in a rough approximation to the desired inversion. Thus, even with no subsequent image distortions, the detector will not extract s, but rather an approximation, $v = \hat{s}$. If this vector is not similar enough to the watermark, w (as measured by the correlation coefficient), to result in a detection, the embedder can be applied again. After each application of the embedder, the extracted vector, v, moves closer to w and the detection value increases. We have found that three or four iterations usually suffice to produce an approximation that can be robustly detected.

C. Orientation of Image Boundaries

It is well known that the rectangular boundary of an image usually causes a "cross" artifact in the image's energy spectrum (see Fig. 2). This happens because there is usually a large discontinuity at each edge of the image due to the implicit tiling.

[3] Alliney [31] presents a technique for the efficient direct computation of the polar Fourier transform of an image.

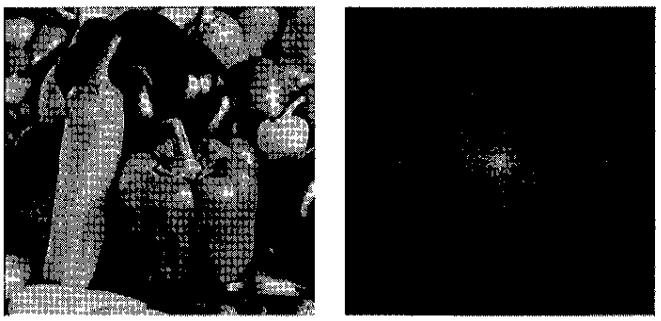


Fig. 2. An image and its DFT.

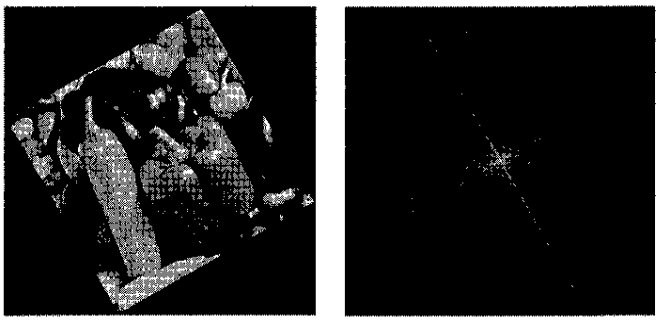


Fig. 3. DFT effects of rotation.

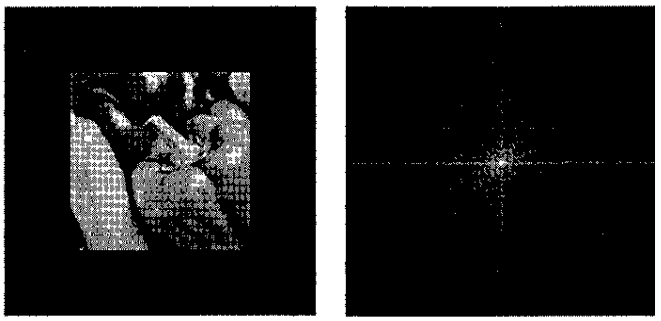


Fig. 4. DFT effects of rotation and cropping.

The DFT magnitude of such vertical and horizontal discontinuities has large energy in all the vertically and horizontally oriented frequencies, which results in the cross artifact.

If the image is rotated, but padded with black so that no image content is cropped, then the cross in the DFT magnitude will also rotate (Fig. 3). If, on the other hand, the rotated image is cropped, so that no black is added, then the new image boundaries cause a horizontal and vertical cross similar to that found in the original image, even though the rest of the DFT magnitude is rotated (Fig. 4). Since the cross has so much energy, it tends to cause two large bumps in the extracted watermark vector, which substantially reduce the correlation coefficient with the embedded watermark.

Our present solution to this problem is to simply ignore the bumps in the extracted signal by ignoring a neighborhood around each of the two highest-valued elements. Alternative solutions that appear in the literature include multiplication of the image by a circularly symmetric window [32] and blurring of the image edges [33]. These solutions are probably more general than the one employed here, but would require modification to the watermark embedder, and have been left for future work.

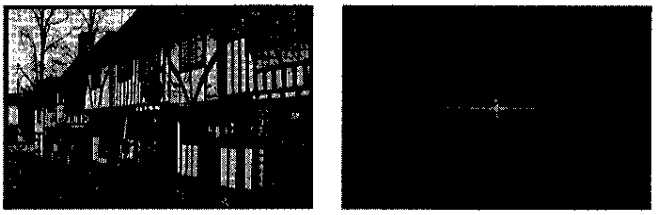


Fig. 5. Image with dominant vertical structure and its DFT.

D. Dynamic Range of Frequency Magnitudes

The magnitude of low frequencies can be very much larger than the magnitude of mid and high frequencies. In these circumstances, the low frequencies can become overwhelming. To reduce this problem, we sum the logs of the magnitudes of the frequencies along the columns of the log-polar Fourier transform, rather than summing the magnitudes themselves.

A beneficial side-effect of this is that a desired change in a given frequency is expressed as a fraction of the frequency's current magnitude rather than as an absolute value. This is better from a fidelity perspective.

E. Unreliability of Extreme Frequencies

It is well known that the lowest and highest frequencies in an image are usually unreliable for watermarking. The low frequencies are unreliable because they are difficult to modify without making visible changes in the image. The high frequencies are unreliable because they can be easily modified by common processes such as compression, printing, and analog transmission. Our solution is to neglect these unreliable frequencies when extracting the watermark.

A better solution would be to use a perceptual model to estimate the maximum amount of change that can be applied to each frequency and a model of specific attacks to estimate the degree of robustness. The amount of watermark energy embedded into each frequency would then be proportional to this perceptual significance and robustness. Such an approach is discussed in [3], [34]–[36]. Application of this idea to the present watermarking method is a topic for future research.

F. Images are Rotationally Asymmetric

The energy in an image is seldom evenly distributed in angular frequency. Images frequently have a large amount of energy in one group of directions, while having much lower energy in an orthogonal group of directions. For example, images containing buildings and trees have significant vertical structure yielding more energy in the horizontal frequencies than in the vertical (Fig. 5), while seascapes or sunsets are strongly oriented in the horizontal direction yielding higher vertical frequencies (Fig. 6).

Spectra such as those of Figs. 5 and 6 suggest an uneven masking ability in orthogonal directions. As a consequence, it may be much easier, from a fidelity perspective, to embed some portions of the watermark than others. For example, when watermarking the image of tall buildings, we can more easily hide

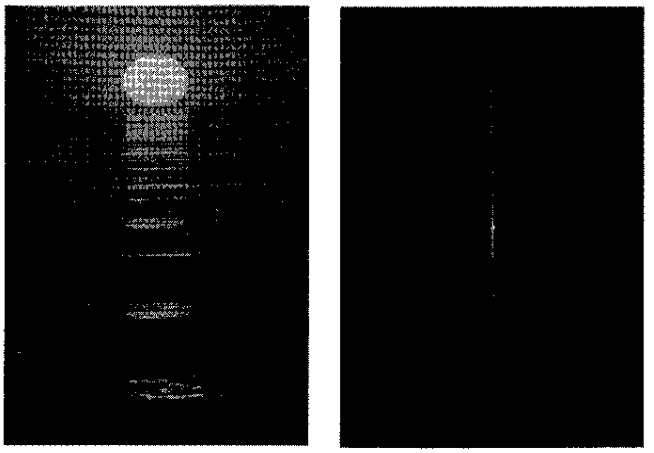


Fig. 6. Image with dominant horizontal structure and its DFT.

noise with a strong vertical component than noise with a strong horizontal component. This can be a problem if the difficult-to-modify portions of the watermark are critical in differentiating it from other watermarks.

To reduce this problem, we divide the extracted signal into two halves, and add the two halves together. Thus, rather than using $g(\theta)$ of (10), we use $g_1(\theta)$ of (12).

If we want to modify an element of $g_1(\theta)$, we can do so by hiding noise that's oriented along either angle θ or angle $\theta+90°$. This increases the likelihood that each element of the watermark can be embedded within the fidelity constraints.

G. High Correlation between Elements of Extracted Watermark

For natural images, $g_1(\theta)$ is likely to vary smoothly as a function of θ. In other words, the extracted signal will have more low-frequency content than high-frequency content. This reduces the effectiveness of the correlation coefficient as a detection measure.

We improve the detection measure by applying a whitening filter to both the extracted signal and the watermark being tested for before computing the correlation coefficient. Note that the whitening filter is employed only in the watermark detector; the embedder is unchanged. The whitening filter was designed to decorrelate the elements of signals extracted from natural images, and was derived from signals extracted from 10 000 images from [37]. These images were not used in any of the subsequent experiments reported in Section IV.

The idea of using a whitening filter to improve watermark detection in this way has been discussed in [38].

H. Interrelation between Changes made in Watermark Elements

During watermark embedding, it is difficult to change the value of one element of the extracted watermark, without changing the values of its neighbors. This results primarily from the fact that any one frequency in the DFT can effect several values of $g_1(\theta)$, so changing that frequency can effect several elements of the watermark. Because of this, it is difficult to embed a watermark that varies wildly from one element to the next.

We reduce this problem by replicating elements of the desired watermark to obtain a lower-frequency watermark. For example, if the watermarks are extracted by computing 74 samples of $g_1(\theta)$ (after removing the samples that contain the "bumps" discussed in Section III-C), then we would define our desired watermark as a vector of 37 values, and duplicate each of its 37 values to obtain a length 74 vector.

IV. Experimental Results

The following results were obtained by extracting a length 90 vector from the image and neglecting the 16 samples surrounding the peak (assumed to correspond to the DFT cross artifact). This leaves a descriptor that is 74 samples in length. The detection process involves a comparison of the watermark with all 90 cyclic rotations of the extracted descriptor. In this section we examine the false positive behavior, effectiveness, and robustness of the proposed scheme. False positive measurements were collected on 10 000 unwatermarked images,[4] and effectiveness and robustness measurements were collected on 2000 watermarked images.

A. Probability of False Positive

We begin our evaluation of the new watermarking method by finding the relationship between the threshold and the probability of false positive. A false positive or false detection occurs when the detector incorrectly concludes that an unwatermarked image contains a given watermark. Thus, the probability of false positive is defined as

$$P_{fp} = P\{D_{\max} > T\} \tag{17}$$

where $D_{\max}$ is a detection value obtained by running the detector on a randomly selected, unwatermarked image and T is the detection threshold. The subscript max specifies the maximum detection value from all of the cyclical shifts examined.

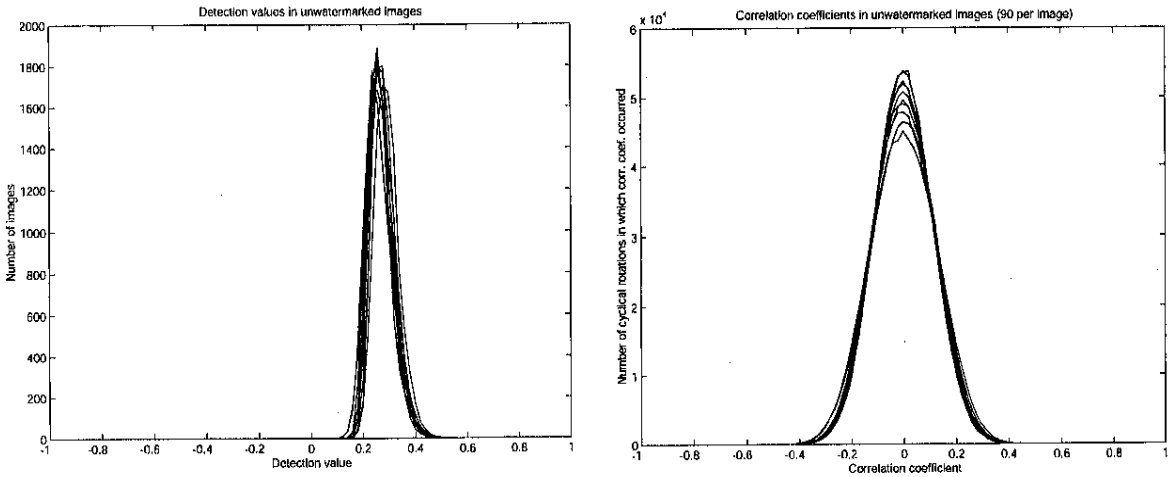
This probability is estimated empirically by applying the detector to 10 000 unwatermarked images from [37], testing for 10 different binary watermarks in each. The ten resulting histograms are shown in Fig. 7(a) superimposed on one another. The probability of false positive is then plotted in Fig. 8(b) as a function of threshold. Again, each trace corresponds to one of the ten watermarks.

Fig. 7(a) indicates that most detection values from unwatermarked images fall between 0.2 and 0.4. This might seem surprising, since we might expect unwatermarked images to yield detection values closer to zero. The reason the values are so high is that each one is the maximum of 90 different correlation coefficients, computed during the cyclical search (see Section II-A, step 3). This means that

$$\begin{aligned} P_{fp} &= P\{D_{\max} > T\} \\ &= P\{(D_0 > T) \text{ or } (D_1 > T) \text{ or } \cdots (D_{89} > T)\} \end{aligned} \tag{18}$$

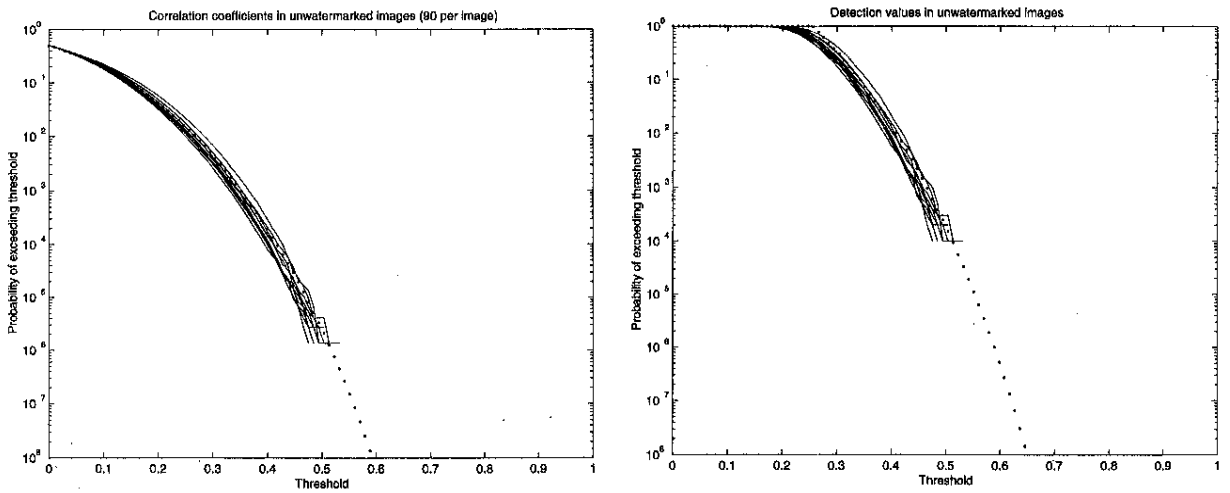
where $D_0 \cdots D_{89}$ are the 90 correlation coefficients computed during the search. Each of $D_0 \cdots D_{89}$ is drawn from a distribution that is centered around zero, as shown in Fig. 7(b), which

[4]The images used in this test were all different from, but from the same database as the 10 000 images that were used to generate the whitening filter.



(a) (b)

Fig. 7. Detection value distributions for ten watermarks in 10 000 unwatermarked images: (a) maximum detection value for each watermark/image pair and (b) all 90 detection values for each watermark/image pair.



(a) (b)

Fig. 8. False positive rates measured with 10 000 unwatermarked images: (a) individual correlation coefficients and (b) final detection value. Each solid trace corresponds to one of ten different watermark vectors. The dotted line represents theoretical estimates.

shows ten superimposed histograms of the 90 000 correlation coefficients computed for each of the 10 watermarks during the experiment. The maximum of 90 values drawn from a distribution like that in Fig. 7(b) is likely to be higher than zero.

During the experiment with unwatermarked images, the highest detection value obtained was 0.55. Thus, we have no data to estimate P_{fp} for $T > 0.55$. To estimate this, we must employ a theoretical model, such as the one described in [39]. This model says that, if D is the correlation coefficient between a preselected d-dimensional watermark vector and a random vector drawn from a radially-symmetric distribution, then

$$P\{D > T\} = R(T,d) = \frac{\displaystyle\int_0^{\cos^{-1}(T)} \sin^{d-2}(u)\, du}{\displaystyle 2\int_0^{\pi/2} \sin^{d-2}(u)\, du}. \quad (19)$$

The whitening filter employed in our detector makes the distribution roughly spherical, so this model is expected to apply to the present system, with $d = 74$. The resulting false positive prediction is shown as a dotted line in Fig. 8(a). The model predicts the probability that one correlation coefficient is greater than the threshold, not the probability that the maximum of several coefficients is greater. Thus, it predicts $P\{D_i > T\}, i \in [0 \cdots 89]$, rather than $P\{D_{\max} > T\}$. Fig. 8(a) indicates how well the model predicted $P\{D_i > T\}$ in our experiment.

We obtain an estimated upper bound on $P\{D_{\max} > T\}$ by observing that

$$P\{Q_0 \text{ or } Q_1 \text{ or } \cdots Q_{n-1}\} \le \min\left(1, \sum_i P\{Q_i\}\right). \quad (20)$$

When Q_i corresponds to the event $(D_i > T)$, and $n = 90$, we obtain

$$P\{D_{\max} > T\} \le \min(1, 90 \times R(T, 74)). \quad (21)$$

This prediction is shown in Fig. 8(b) as a dotted line.

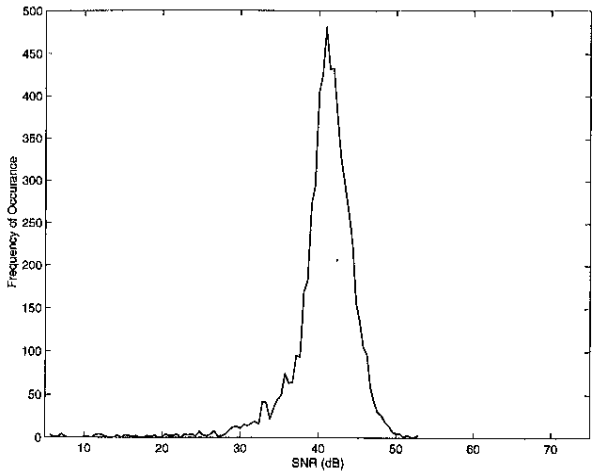


Fig. 9. Signal-to-noise ratio.

Fig. 10. Watermarking with little impact on fidelity.

B. Fidelity

The tradeoff between fidelity and robustness is controlled by adjusting the relative weighting used in the mixing of the watermark signal and the signal extracted from the original image (see Section II-B). As the relative weight assigned to the watermark signal is increased, the strength of the embedded watermark is increased at the expense of lower fidelity. Once chosen, the mixing weights were held constant over all experiments described in this section. These weights were empirically selected to yield an average signal-to-noise ratio of about 40 dB.[5] Fig. 9 shows a histogram of the ratios obtained. Fig. 10 shows an example of a watermarked image with little impact on fidelity.

It must be noted, however, that signal-to-noise ratio is not a very effective predictor of perceptual quality. The fidelity of the image depends to a large degree on the perceptual relationship between the image and the noise. In general, noise that matches the underlying textures in an image is less perceptible than noise that is very different from the image, even at the same signal-to-noise ratios.

The present system generates watermark patterns by making small percentage adjustments to the powers of frequencies in the image's spectrum, so the resulting noise pattern is usually similar to the textures in the image. Thus, when we watermark an image that contains a homogeneous texture, the watermark is well-hidden. But when we mark an image that contains widely varying textures, the mark can become visible. Fig. 11 illustrates the problem. The watermark strength in this figure was increased so that the problem should be visible after printing in this TRANSACTIONS.

[5]Here the "signal" is the image, and the "noise" is the watermark pattern.

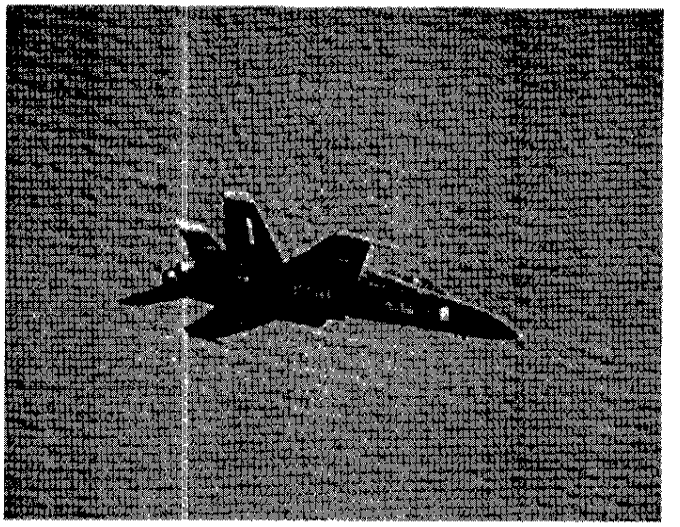


Fig. 11. Character of the watermark noise when the strength is too high. The watermark strength in this figure was increased so that the problem should be visible after printing in this TRANSACTIONS.

Solving the fidelity problem in nonhomogeneous images would require a modification to the algorithm that attenuates or shapes the watermark according to local texture characteristics. This has been left for future work.

C. Effectiveness

The effectiveness of a watermarking scheme is measured as the probability that the output of the watermark embedder will contain the watermark, subject to constraints on the fidelity of the marked image and the detection threshold or probability of false detection. In other words, effectiveness is the probability of true detection when the marked image is not subjected to any distortions after embedding. The effectiveness of the current scheme is measured and plotted as the dashed ROC curves[6] in each of Figs. 13–20.

D. Robustness

In a practical setting, RST distortions are usually accompanied by cropping. Fig. 12(f), (g), and (i) show respectively rotation, scaling, and translation with the associated cropping. With the current algorithm, cropping can be viewed as distortion of the extracted signal by additive noise. As such, we expect cropping to degrade the detection value.

In this section, seven geometric distortion attacks are examined: rotation with and without cropping, scaling up with and without cropping, translation with and without cropping, and scaling down. Note that scaling down does not imply cropping.

[6]A receiver-operating-characteristic (ROC) curve is a plot of the probability of true detection as a function of the probability of false detection. For details see [40].

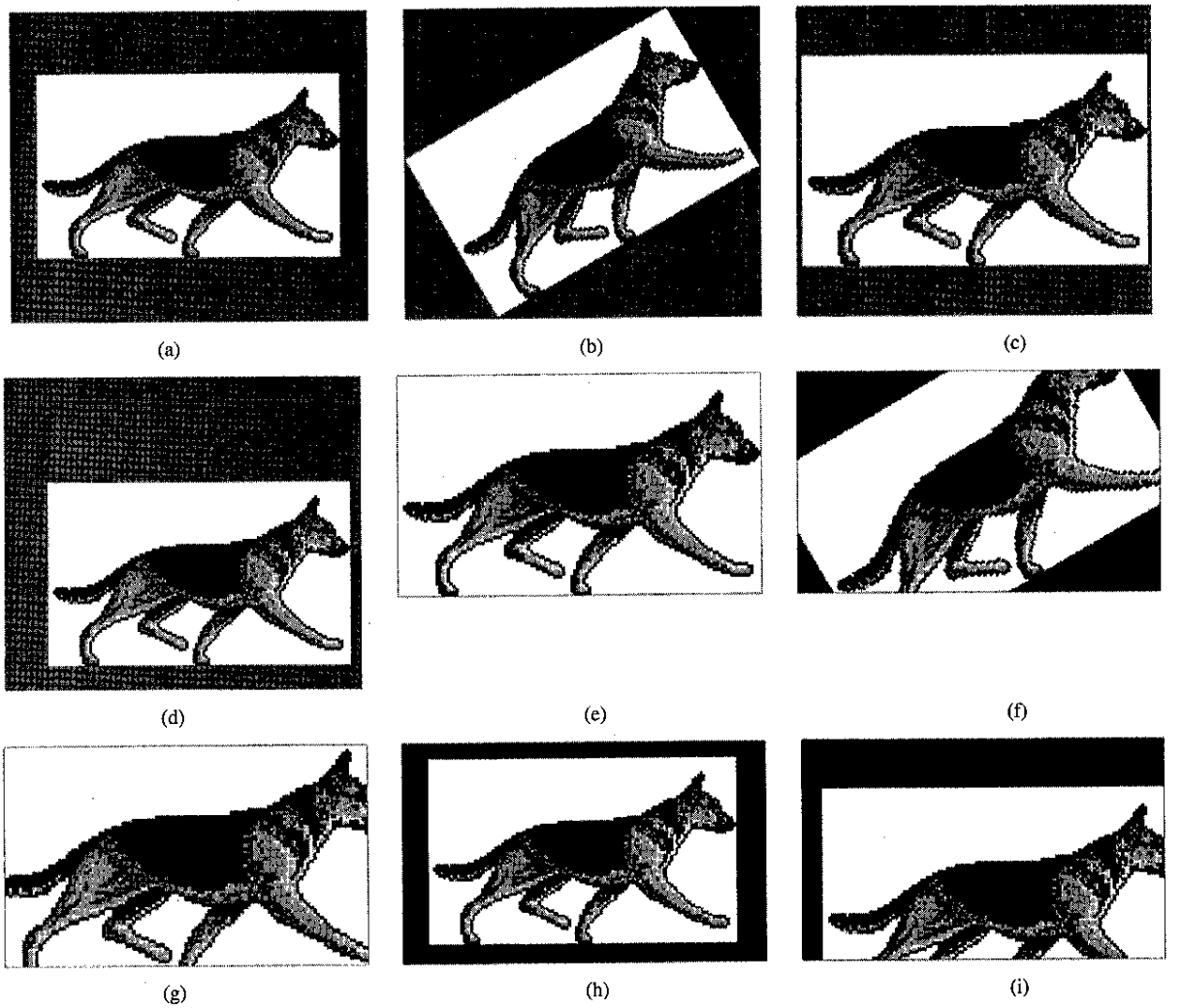


Fig. 12. Examples of geometric attacks: (e) and (a) are the original and padded original respectively; (b)–(d) attacks without cropping; and (f)–(i) attacks with cropping.

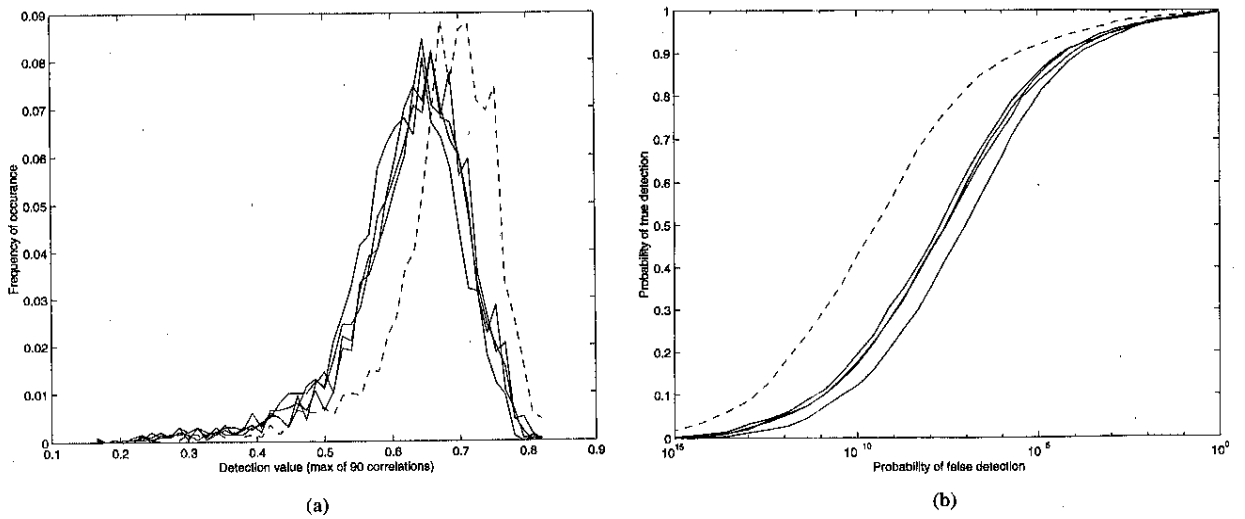
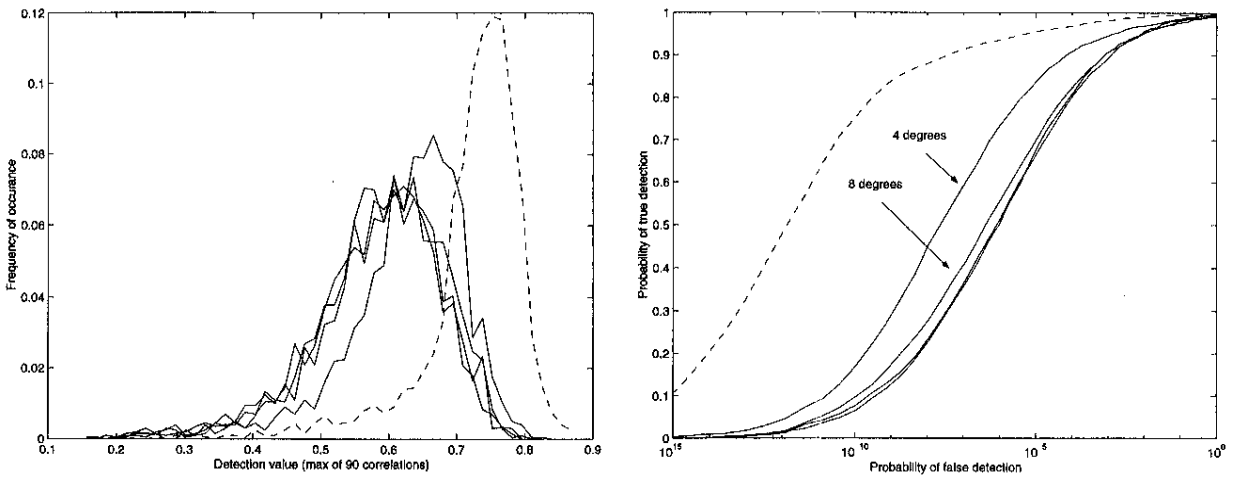


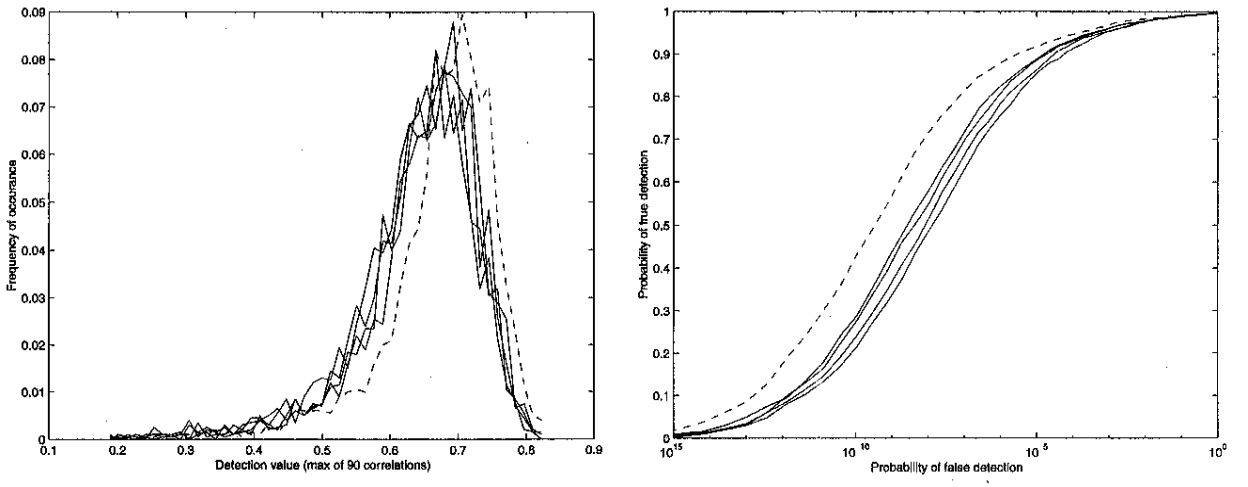
Fig. 13. Rotation without cropping, 4°, 8°, 30°, and 45°. (a) Histogram and (b) ROC (dashed line represents the detection values prior to attack).

In order to isolate the effects of rotation, scaling up, and translation from cropping, the images have been padded with gray as shown in Fig. 12(a). The embedder has been applied to these expanded images and then the gray padding replaced with unwatermarked gray padding prior to detection or attack. The amount of padding is such that none of the rotation, scaling up, and translation experiments cause image data to be cropped. The only data that is cropped is unwatermarked padding. Thus, the



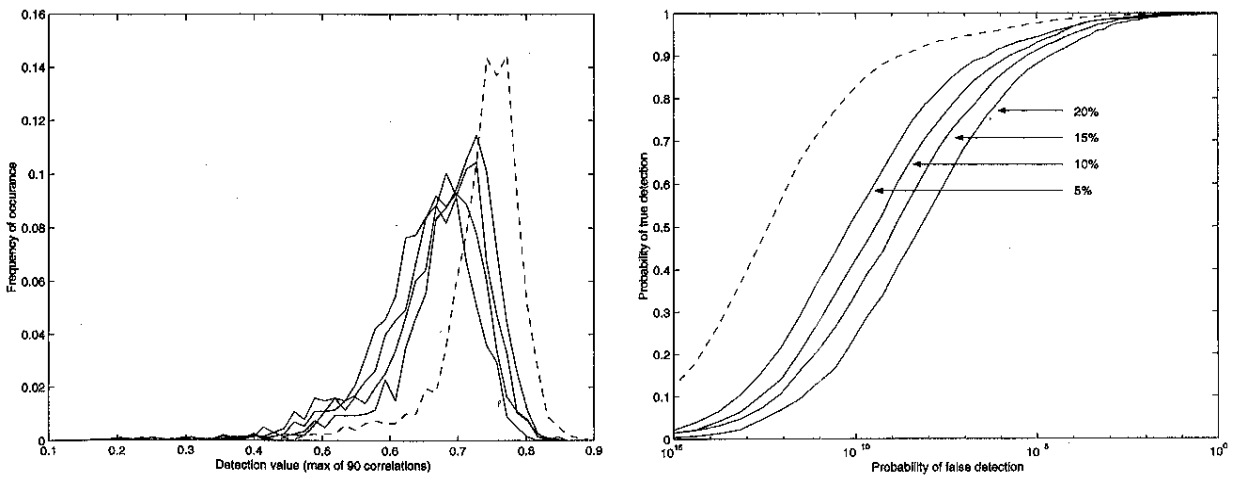
(a) (b)

Fig. 14. Rotation with cropping, 4°, 8°, 30°, and 45°. (a) Histogram and (b) ROC (dashed line represents the detection values prior to attack).



(a) (b)

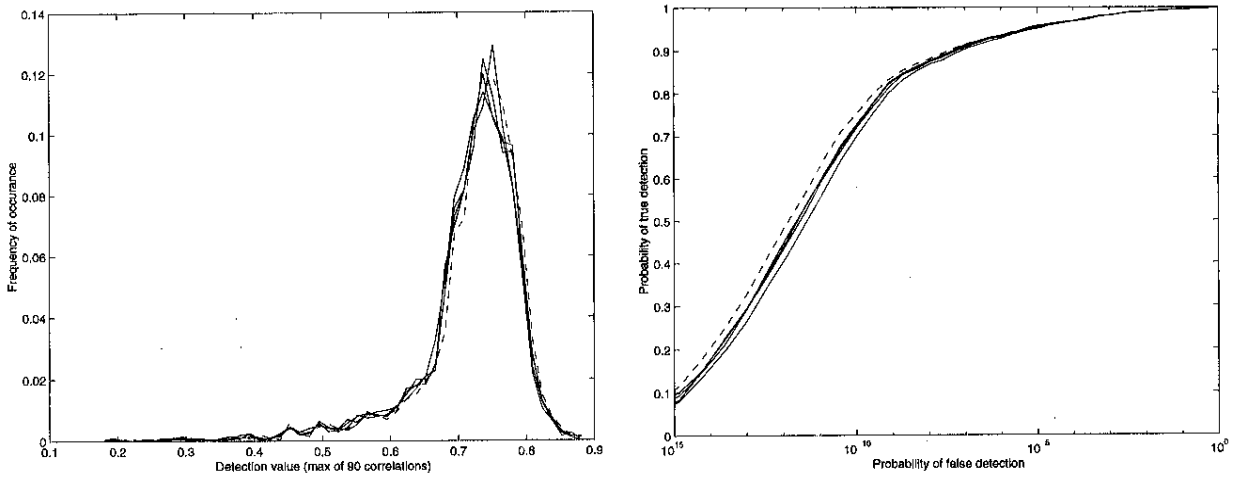
Fig. 15. Scaling up without cropping, 5%, 10%, 15%, and 20%. (a) Histogram and (b) ROC (dashed line represents the detection values prior to attack).



(a) (b)

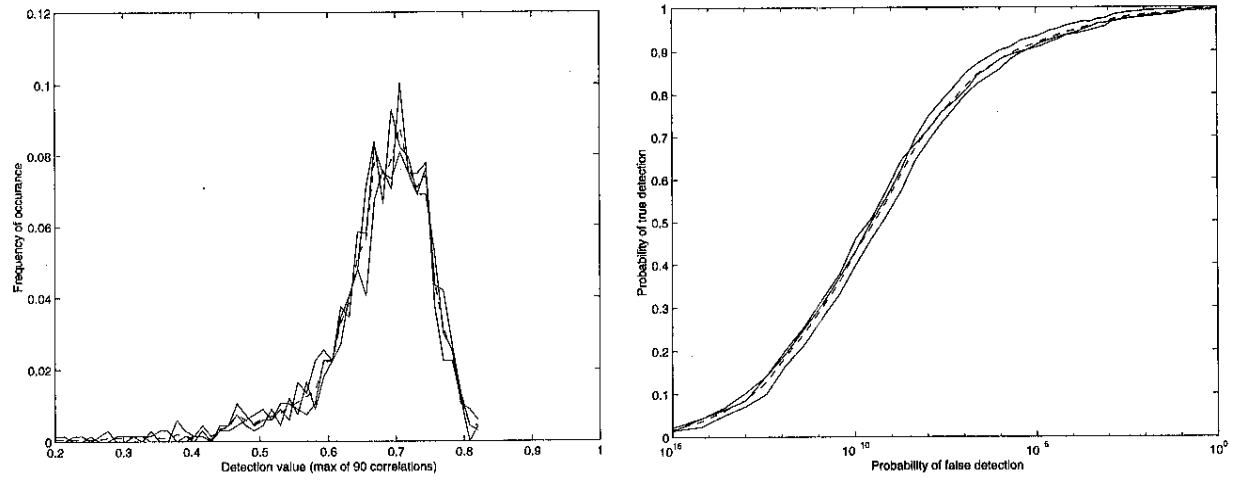
Fig. 16. Scaling up with cropping, 5%, 10%, 15%, and 20%. (a) Histogram and (b) ROC (dashed line represents the detection values prior to attack).

differences between the detection values prior to rotation and those after rotation can be attributed solely to the rotation as the associated cropping of unwatermarked padding does not effect the detection value.



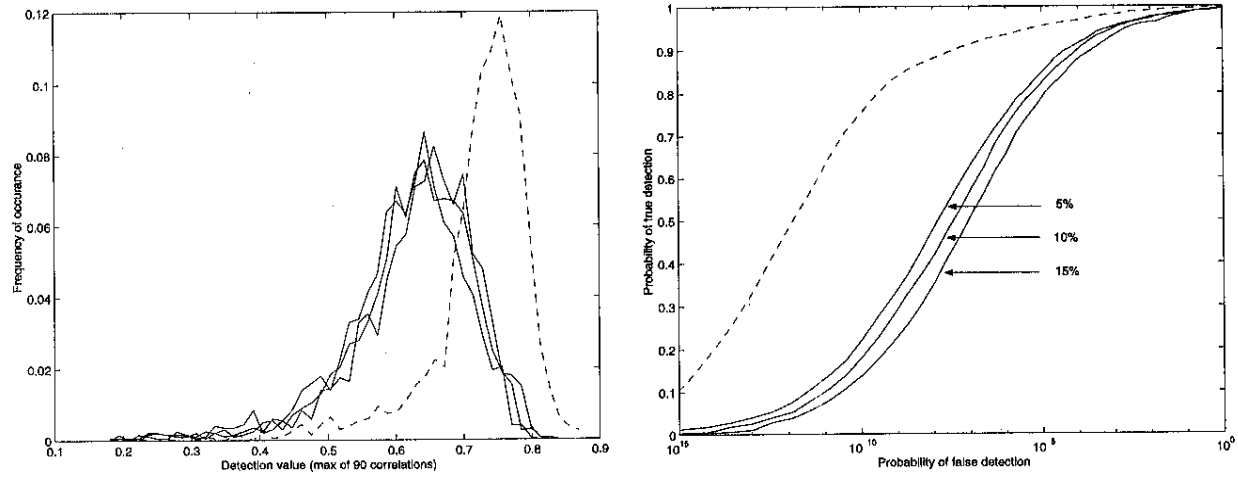
(a) (b)

Fig. 17. Scaling down, 5%, 10%, 15%, and 20%. (a) Histogram and (b) ROC (dashed line represents the detection values prior to attack).



(a) (b)

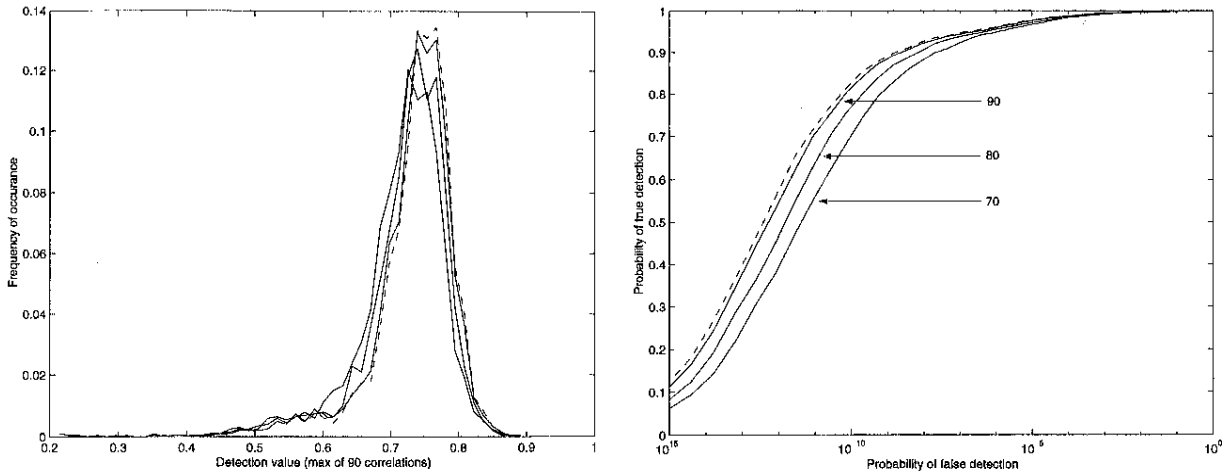
Fig. 18. Translation without cropping, 5%, 10%, and 15%. (a) Histogram and (b) ROC (dashed line represents the detection values prior to attack).



(a) (b)

Fig. 19. Translation with cropping, 5%, 10%, and 15%. (a) Histogram and (b) ROC (dashed line represents the detection values prior to attack).

The detection value prior to attack is used to measure the effectiveness of the watermarking scheme. This effectiveness is likely to be reduced in the padded examples since a portion of the watermarked image (the watermarked gray padding) has been replaced with nonwatermarked padding. However, the purpose of the experiments based on these padded geometric at-



(a) (b)

Fig. 20. JPEG compression, $QF = 90, 80$, and 70. (a) Histogram and (b) ROC (dashed line represents the detection values prior to attack).

tacks, shown in Fig. 12(b)–(d), is to isolate the effects due to geometric distortions from those due to cropping.

1) Rotation: Two experiments were performed to test the watermark's robustness against rotation. The first experiment was designed to isolate the effects of rotation from all other types of attack. The second was a more realistic test of the effects of rotation with cropping.

Each trial of the first test comprised the following steps.

1) Pad an image with neutral gray, increasing its size. The amount of padding was chosen to allow rotation without any part of the original image going outside of the image boundaries [Fig. 12(a)].
2) Embed a randomly-selected watermark in the padded image.
3) Replace the padding with neutral gray again. This removes any watermark information from the neutral gray area.
4) Run the watermark detector on the image to obtain a detection value before rotation.
5) Rotate the image by a predetermined angle, and crop to the original size. Fig. 12(b) shows what an image looks like after this step. Note that only the padding is cropped, so we do not crop off any of the watermark pattern.
6) Run the watermark detector on the image to obtain a detection value after rotation.

Since the padding that's cropped off during rotation contains no watermark pattern, any difference between the "before" value obtained in step 4 and the "after" value obtained in step 6 can only result from the effects of rotation.

This experiment was performed on 2000 images with rotations of 4°, 8°, 30°, and 45°. We limited this test to a maximum rotation of 45° because rotations beyond 45° are equivalent to smaller rotations after a rotation of 90°. An image that has been rotated 90° yields exactly the same extracted vector as an unrotated image, so a rotation of greater than 45° should behave the same as a smaller rotation.

As indicated in Fig. 13(a), the different rotations yielded essentially the same results. Fig. 13(b) shows ROC curves before and after rotation. For each of the ROC curves, the false-positive probabilities were estimated using the method described in Section IV-A. In the two plots of Fig. 13, the dashed lines represent the detection values prior to attack, i.e., the effectiveness of the embedding. The deviations of the solid traces from the dashed represent the effects of the attack.

In the second experiment, we watermarked the original image without padding, and allowed part of the watermark pattern to be cropped off after rotation. Fig. 12(f) shows an example of what an image looked like after the rotation. This experiment was performed on 2000 images with rotations of 4°, 8°, 30°, and 45°. Fig. 14 shows the results.

Three immediate observations based on the ROC curve of Fig. 13(b) are that the effects of these four rotations are all similar, for a fixed false positive probability, P_{fp}, (independent axis) rotation decreases the likelihood of detection (difference between the dashed and solid lines), and the effect of rotation on the probability of detection is dependent on the P_{fp} or equivalently the threshold. For relatively high P_{fp}, for example 10^{-3} or one in a thousand, the current method is extremely robust to rotation. At smaller values of P_{fp}, for example 10^{-8}, rotation degrades the detection value more significantly. Fig. 14(b) further shows that the cropping that accompanies rotation has a significant, negative impact on detection [downward shift of the solid lines in Fig. 14(b) from those in Fig. 13(b)], and the deterioration of the detection value is more dependent on rotation angle (different rotations result in different amounts of cropping).

These ROC curves emphasize the importance of the baseline measurement (dashed lines), which serves as an upper bound on robustness. They also show that each of the two experiments begin from a different baseline. In the second experiment, the rotation attack is applied to images that have been much more effectively watermarked. The lower effectiveness of the first experiment represents the cropping of watermarked data that occurs when the watermarked gray padding is replaced with unwatermarked gray padding. Recall that these somewhat artificial embedding conditions are in place to isolate the effects of rotation from any further degradation that may occur due to the cropping that normally accompanies rotation.

These results demonstrate that the current watermark, designed to be invariant to rotations, does exhibit a resilience to rotation. This watermark has not been explicitly designed to withstand cropping and the results highlight this fact.

2) Scale: To test robustness to scaling, we performed three experiments. The first and second test the effect of scaling up, with and without cropping. The third tests the effect of scaling down, with padding.

In the first scaling test, the steps performed for each trial were the same as those for the first rotation step, with the exception that instead of rotating the image we scaled the image up. Fig. 12(c) shows an example of an image that has been scaled up after padding and watermarking. The test was performed on 2000 images at scales 5%, 10%, 15%, and 20% larger than the original. The results are shown in Fig. 15.

The second test was the same as the first except without padding the images before scaling, so part of the image was cropped off after scaling. Fig. 12(g) illustrates the attack. The test was performed on 947 images at scales of 5%, 10%, 15%, and 20% larger than the original. The results are shown in Fig. 16.

For the test of reduced scaling, we do not have to be concerned with cropping. After watermarking and scaling, the image is padded back to its original size. This padding is part of the scaling attack, not an operation performed by the detector. Since cropping is not an issue here, the image need not be padded with gray before watermarking and we performed only one version of the experiment. The scale-down attack is illustrated in Fig. 12(h). The test was performed on 2000 images at scales 5%, 10%, 15%, and 20% smaller than the original. The results are shown in Fig. 17.

As with rotation, the results show that scaling up, in general, degrades the probability of detection as a function of P_{fp}. For the relatively high $P_{fp} = 10^{-3}$, scaling has very little effect on the likelihood of detection while at $P_{fp} = 10^{-8}$ the effect is more significant. We also observe that the results differ slightly for different scale factors at these lower false positive rates.

The differences between the ROC curves in Figs. 15 and 16 clearly show the severe degradation due to the cropping that normally accompanies scaling. As expected, the effect of this cropping increases with the scale factor because higher scale factors imply more cropping.

Fig. 17 shows that a decrease in scale has virtually no effect for $P_{fp} > 10^{-7}$ or so and for lower P_{fp} the degradation is only slight.

The current watermark was designed to be invariant to changes in scale and these results demonstrate an excellent resilience to a decrease in scale and good resilience to an increase in scale. Again, these results highlight the negative impact of cropping.

3) Translation: We expect translation alone to have no effect on the watermark, since the watermark is computed from the magnitudes of the Fourier coefficients. To test this, we performed two experiments.

The first experiment was similar to the first rotation and scaling experiments, in that the image was padded before watermarking and the padding was replaced after watermarking. We then translated the image by cropping gray off the bottom and right, and padding gray onto the top and left. Fig. 12(d) shows an example of such a translated image. The experiment was performed on 2000 images at translations of 5%, 10%, and 15% of the image size. The results are shown in Fig. 18.

The second translation test was performed without padding the image before translation, so that part of the watermark pattern is cropped during translation. Fig. 12(i) shows an example of this attack. Again, the experiment was performed on 2000 images at translations of 5%, 10%, and 15% of the image size. The results are shown in Fig. 19.

The results of the first experiment show that translation has negligible effect on probability of detection. This means that the second test is more a test of robustness to cropping than to translation, and we see the same sort of pattern that was observed in the second rotation and scaling experiments.

E. JPEG Compression

While the purpose of the present watermark design is to survive RST transformations, it may be important that the watermarks also survive other common types of image processing. We therefore conducted a test of robustness to JPEG compression.

After watermarking, images were JPEG compressed at quality factors of 90, 80, and 70 using Equilibrium Debabelizer Pro. The test was performed on 2000 images. Fig. 20 shows the results.

The results show that the likelihood of detection decreases with the amount of compression noise introduced and that this decrease is dependent on the P_{fp}. For relatively high $P_{fp} = 10^{-3}$, the current method is extremely robust to JPEG compression at the qualities tested. At more restrictive false positive probabilities, for example 10^{-8}, JPEG at $QF = 70$ still yields a respectable robustness of about 88%.

V. Conclusion

Geometric distortions continue to be a major weakness for many watermarking methods. We described a solution to the common problems of rotation, scale, and translation. This solution is related to earlier proposals in the pattern recognition literature regarding invariants of the Fourier–Mellin transform. However, unlike those proposals, we do not explicitly derive an invariance relationship.

Instead of creating a truly RST invariant signal, we create a signal that changes in a trivial manner as a result of rotation, scale, or translation. The calculation of this projection is performed by taking the Fourier transform of the image, performing a log-polar resampling, and then integrating along the radial dimension. We note that an alternative implementation can be performed using the Radon transform [27]. We have investigated this implementation but do not report it here.

The 1-D watermark has a many-to-one mapping to the 2-D image space. This is advantageous, especially when the embedder is based on the principle of communications with side information. Our implementation is a very simple example of this principle and we believe that future work can lead to significant improvements.

Experimental results on a database of over 2000 images demonstrate that the method is resilient to either rotations, scale changes, or translations. In addition, the technique is resilient to mild JPEG compression. The degree of resilience changes as a function of the probability of false positive. The results also demonstrate the weakness of this method to cropping, an attack against which no steps have been taken in the design.

Future work will focus on more effective embedding and RST resilient watermarking designed to survive cropping and compression. Improvements in effectiveness are possible in the approximate inversion of the log-polar resampling and in the distribution of the difference signal to the log-polar coefficients. Methods based on gradient descent will be investigated. Also, the current technique of uniform distribution does not fully exploit the visual properties of the host image.

We will examine techniques for building crop resistant watermarks that rely on first subdividing the image into a number of possibly overlapping, tiles. The RST resilient watermark is then embedded in each of these tiles. The detection algorithm is applied to each tile and the results averaged together. With appropriate constraints on the tiling and the symmetry of the watermark this technique may provide the desired resilience to cropping.

ACKNOWLEDGMENT

The authors would like to thank Dr. H. Stone of NEC Research Institute for helpful discussions on image registration and Fourier–Mellin transform. In addition, the authors would like to thank Dr. J. J. K. O'Ruanaidh, Siemens Research Center, Princeton, NJ, for useful discussions. The test images used in experiments are from the Corel Stock Photo Library 3 image database [37].

REFERENCES

[1] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Attacks on copyright marking systems," in *Proc. Workshop Information Hiding*, Portland, OR, Apr. 15–17, 1998.

[2] *Unzign*, http://altern.org/watermark.

[3] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Processing*, vol. 6, pp. 1673–1687, Dec. 1997.

[4] N. F. Johson, Z. Duric, and S. Jajodia, "Recovery of watermarks from distorted images," in *Proc. 3rd Int. Information Hiding Workshop*, 1999, pp. 361–375.

[5] C.-Y. Lin and S.-F. Chang, "Distortion modeling and invariant extraction for digital image print-and-scan process," in *Int. Symp. Multimedia Information Processing*, 1999.

[6] S. Pereira and T. Pun, "Fast robust template matching for affine resistant image watermarks," in *Proc. 3rd Int. Information Hiding Workshop*, 1999, pp. 207–218.

[7] G. Csurka, F. Deguillaume, J. J. K. O'Rauanaidh, and T. Pun, "A Bayesian approach to affine transformation resistant image and video watermarking," in *Proc. 3rd Int. Information Hiding Workshop*, 1999, pp. 315–330.

[8] C. Honsinger and M. Rabbani, "Data embedding using phase dispersion," in *PICS 2000: Image Processing, Image Quality, Image Capture, Systems Conf.*, vol. 3, 2000, pp. 264–268.

[9] C. W. Honsinger and S. J. Daly, "Method for detecting rotation and magnification in images," U.S. Patent, 5 835 639, 1998.

[10] M. Kutter, "Watermarking resistance to translation, rotation, and scaling," *Proc. SPIE Multimedia Systems Applications*, vol. 3528, pp. 423–431, 1998.

[11] M. Alghoniemy and A. H. Tewfik, "Geometric distortion correction through image normalization," in *Proc. Int. Conf. Multimedia Expo*, vol. 3, 2000, pp. 1291–1294.

[12] M. K. Hu, "Visual pattern recognition by moment invariants," *IEEE Trans. Inform. Theory*, vol. IT-8, pp. 179–187, 1962.

[13] Y. Sheng and H. H. Arsenault, "Experiments on pattern recognition using invariant Fourier–Mellin descriptors," *J. Opt. Soc. Amer. A*, vol. 3, no. 6, pp. 771–776, 1986.

[14] D. Casasent and D. Psaltis, "Position, rotation, and scale invariant optical correlation," *Appl. Opt.*, vol. 15, no. 7, pp. 1795–1799, 1976.

[15] ——, "New optical transforms for pattern recognition," *Proc. IEEE*, vol. 65, pp. 77–84, Jan. 1977.

[16] J. Altmann and H. J. P. Reitbock, "A fast correlation method for scale- and translation-invariant pattern recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 46–57, Jan. 1984.

[17] J. Altmann, "On the digital implementation of the rotation-invariant Fourier–Mellin transform," *J. Inform. Process. Cybern.*, vol. EIK 28, pp. 13–36, 1987.

[18] H. Wechsler and G. L. Zimmerman, "2-D invariant object recognition using distributed associative memory," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 6, pp. 811–821, 1988.

[19] F. Lin and R. D. Brandt, "Toward absolute invariants of images under translation, rotation, and dilation," *Pattern Recognit. Lett.*, vol. 14, no. 5, pp. 369–379, 1993.

[20] M. Ferraro and R. M. Caelli, "Lie transform groups, integral transforms, and invariant pattern recognition," *Spatial Vis.*, vol. 8, no. 1, pp. 33–44, 1994.

[21] J. J. K. O'Ruanaidh and T. Pun, "Rotation, scale and translation invariant spread spectrum digital image watermarking," *Signal Process.*, vol. 66, no. 3, pp. 303–317, 1998.

[22] I. J. Cox, M. L. Miller, and A. McKellips, "Watermarking as communications with side information," *Proc. IEEE*, vol. 87, pp. 1127–1141, July 1999.

[23] T. Kalker, G. Depovere, J. Haitsma, and M. J. Maes, "Video watermarking system for broadcast monitoring," *Proc. Security Watermarking System Broadcast Monitoring*, pp. 103–112, 1999.

[24] P. Bas, J.-M. Chassery, and F. Davoine, "A geometrical and frequential watermarking scheme using similarities," in *Proc. Security Watermarking System Broadcast Monitoring*, vol. SPIE-3657, 1999, pp. 264–272.

[25] M. J. J. B. Maes and C. W. A. M. van Overveld, "Digital watermarking by geometric warping," in *IEEE Int. Conf. Image Processing*, vol. 2, 1998, pp. 424–426.

[26] P. M. J. Rongen, M. J. J. B. Maes, and C. W. A. M. van Overveld, "Digital image watermarking by salient point modification practical results," in *Proc. Security Watermarking System Broadcast Monitoring*, vol. SPIE-3657, 1999, pp. 273–282.

[27] R. N. Bracewell, *The Fourier Transform and Its Applications*. New York: McGraw-Hill, 1986.

[28] C. E. Shannon, "Channels with side information at the transmitter," *IBM J. Res. Develop.*, pp. 289–293, 1958.

[29] M. L. Miller, J. A. Bloom, and I. J. Cox, "Exploiting detector and image information in watermark embedding," in *IEEE Int. Conf. Image Processing*, vol. 3, Vancouver, BC, Canada, Sept. 2000, pp. 1–4.

[30] H. S. Stone, B. Tao, and M. McGuire, "Analysis of image registration noise due to rotationally dependent aliasing," NEC Res. Inst. Tech. Rep., TR 98-018, 1998.

[31] S. Alliney, "Digital analysis of rotated images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 499–504, 1993.

[32] E. De Castro and C. Morandi, "Registration of translated and rotated images using finite Fourier transforms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 700–703, 1987.

[33] M. McGuire, "An image registration technique for recovering rotation, scale and translation parameters," NEC Res. Inst. Tech. Rep., TR 98-018, 1998.

[34] I. J. Cox and M. L. Miller, "A review of watermarking and the importance of perceptual modeling," *Proc. SPIE*, vol. 3016, pp. 92–99, 1997.

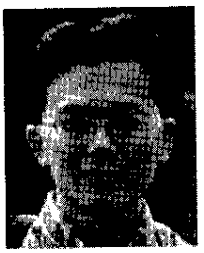
[35] C. I. Podilchuk and W. Zeng, "Image-adaptive watermarking using visual models," *IEEE Trans. Select. Areas Commun.*, vol. 16, pp. 525–539, Apr. 1998.

[36] R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp, "Perceptual watermarks for digital images and video," *Proc. IEEE*, vol. 87, pp. 1108–1126, July 1999.

[37] *Corel Corporation, Corel Stock Photo Library 3.*

[38] G. Depovere, T. Kalker, and J.-P. Linnartz, "Improved watermark detection using filtering before correlation," in *IEEE Int. Conf. Image Processing*, vol. 1, Chicago, IL, Oct. 1998, pp. 430–434.

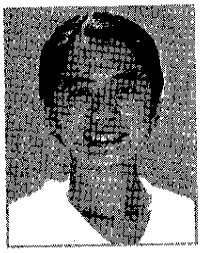
[39] M. L. Miller and J. A. Bloom, "Computing the probability of false watermark detection," in *Proc. 3rd Int. Workshop Information Hiding*, 1999.
[40] H. V. Poor, *An Introduction to Signal Detection and Estimation*. Berlin, Germany: Springer-Verlag, 1994.
[41] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*. San Francisco, CA: Morgan Kaufman, 2001.



Ching-Yung Lin (M'96) received the B.S. and M.S. degrees from National Taiwan University (NTU), Taipei, Taiwan, R.O.C., in 1991 and 1993, respectively, and the Ph.D. degree from Columbia University, New York, in 2000, all in electrical engineering.

From 1993 to 1995, he was a Wireless Communication Engineer with the Air Force, Taiwan. From August 1995 to July 1996, he was an Instructor with the Computer Communication Network Lab, NTU, and a Research Engineer for computer vision system projects with EeRise Co. In October 2000, he joined the IBM T. J. Watson Research Center, Yorktown Heights, NY, as a Research Staff Member. His current research interests include multimedia authentication and watermarking techniques, multimedia indexing and query, multimedia transmission and networking, human-computer Interaction, and multirate multidimensional signal processing. He was the primary contributor in the design of the unique self-authentication-and-recovery image system, which distinguishes JPEG/MPEG compression from malicious manipulation, and in the design of the public/blind watermarking method surviving the print-and-scan process. He has three U.S. patents pending.

Dr. Lin was the recipient of Lung-Teng Thesis Award and an Outstanding Paper Award, both in 1993. He co-chaired Special Sessions on Multimedia Security and Watermarking Applications at IEEE International Conference on Information Technology: Coding and Computing, 2001.

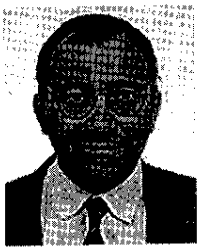


Min Wu (S'95) received the B.E. degree in electrical engineering and the B.A. degree in economics from Tsinghua University, Beijing, China, in 1996 (both with the highest honors), and the M.A. degree in electrical engineering from Princeton University, Princeton, NJ, in 1998, where she is currently pursuing the Ph.D. degree in electrical engineering.

She was with NEC Research Institute and Signafy, Inc., Princeton, in 1998, and with the Media Security Group, Panasonic Information & Networking Laboratories, Princeton, in 1999. Her research interests include digital watermarking and information security, video communication, multimedia signal processing, and content analysis.

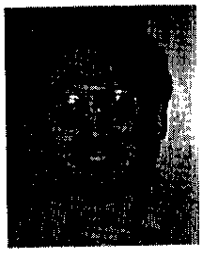
Jeffrey A. Bloom (S'89–M'98) received the B.S. and M.S. degrees from Worcester Polytechnic Institute, Worcester, MA, and the Ph.D. degree in electrical and computer engineering from the University of California, Davis, in 1999.

From 1985 to 1986, he was an Engineer with Eastman Kodak Company working in the Quality Assurance Testing Group. From 1998 to 2000, he was with Signafy, Inc., and then NEC Research Institute, both in Princeton, NJ, as a Research Engineer examining issues related to the watermarking of image, video, and audio data. He is currently a Technology Leader with the Internet and Multimedia Systems Group, Sarnoff Corporation, Princeton. His current research interests include multimedia watermarking, digital rights management, image and video compression, and machine learning.



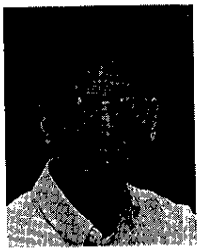
Ingemar J. Cox (S'79–M'83–SM'95) received the B.Sc. degree from University College, London, U.K., and the Ph.D. degree, from Oxford University, Oxford, U.K.

He was Member of Technical Staff with AT&T Bell Labs, Murray Hill, NJ, from 1984 to 1989, where his research interests were focused on mobile robots. In 1989, he joined NEC Research Institute, Princeton, NJ, as a Senior Research Scientist in the Computer Science Division. At NEC, his research shifted to problems in computer vision, and he was responsible for creating the Computer Vision Group at NECI. He has worked on problems to do with stereo and motion correspondence and multimedia issues of image database retrieval and watermarking. From 1997 to 1999, he was Chief Technical Officer with Signafy, Inc., a subsidiary of NEC responsible for the commercialization of watermarking. In 1999, he returned to NEC Research Institute as a Research Fellow. He is on the editorial board of the *International Journal of Autonomous Robots* and co-editor of two books: *Autonomous Robots Vehicles* and *Partitioning Data Sets: With Applications to Psychology, Computer Vision and Target Tracking*.



Matt L. Miller received the B.A. degree in cognitive science from The University of Rochester, Rochester, NY, in 1986.

He began working in computer graphics at AT&T Bell Labs, Murray Hill, NJ, in 1979. He became Lead Programmer at NPS, a startup developing color desktop publishing software. From 1990 to 1993, he delivered graduate-level lecture courses in color graphics at Aarhus University, Denmark, and Charles University and Czech Technical University, Prague, Czech Republic. From 1993 to 1997, he divided his time between running Baltic Images, a company he founded in Lithuania, and consulting for NEC Research Institute, Princeton, NJ. In 1997 he sold Baltic Images, and is now a Research Engineer with NEC Research Institute.



Yui Man Lui (S'96–M'97) was born in Hong Kong in 1967. He received the B.S. degree in information engineering from Feng Chia University, Taichung, Taiwan, R.O.C., in 1991, and the M.S. degree in computer science from Utah State University, Logan, in 1996.

He is currently an Imaging Engineer with Robotic Vision Systems, Inc., Happauge, NY. He has authored and co-authored more than ten refereed journal and international conference papers and has one patent pending. His research interests include mammographic imaging, airglow imaging, face recognition, image watermarking, image registration, industrial visual inspection, and fuzzy logic.

Mr. Lui was the Mathematical Section Winner of Scholar Day at Utah State University in 1996. He also received the best student travel award from the IEEE Nuclear Science and Medical Imaging Conference in 1996.