

RST Invariant Digital Image Watermarking Based on Log-Polar Mapping and Phase Correlation

Dong Zheng, *Student Member, IEEE*, Jiyong Zhao, *Member, IEEE*, and Abdulmotaleb El Saddik, *Senior Member, IEEE*

Abstract—Based on log-polar mapping and phase correlation, this paper presents a novel digital image watermarking scheme that is invariant to rotation, scaling, and translation (RST). We embed watermark in the log-polar mappings of Fourier magnitude spectrum of original image, and use the phase correlation between the LPM of the original image and the LPM of the watermarked image to calculate the displacement of watermark positions in LPM domain. The scheme preserves the image quality by avoiding computing inverse log-polar mapping (ILPM), and produces smaller correlation coefficient for unwatermarked images by using phase correlation to avoid exhaustive search. The evaluations demonstrate that the scheme is invariant to rotation and translation, invariant to scaling when the scale is in a reasonable range, and very robust to JPEG compression.

Keywords—digital image watermarking RST invariant Fourier-Mellin Transform LPM ILPM cross-power spectrum phase correlation

I. INTRODUCTION

The rapid growth of multimedia applications has created an urgent need for adequate copyright schemes especially for image and video data. Robust and practical watermarking techniques can be used to trace copies or to implement copy protection schemes. Researchers have paid great efforts on watermarking techniques, up to now, many watermarking methods have been proposed [1][2][3][4][5].

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. While many methods perform well against compression, they lack robustness to geometric transformations [6]. Rotation and scaling attacks are considered more challenging than other attacks. This is due to the fact that changing the image size or its orientation, even by slight amount, could dramatically reduce the receiver ability to retrieve the watermark [7]. Recently it has been clear that even very small geometric distortions can prevent the detection of a watermark [8]. Lin et al. [8] have given a

Dong Zheng is with Multimedia Communications Lab, School of Information Technology and Engineering, University of Ottawa, 800 King Edward Ave, Ottawa, Ontario, K1N 6N5, Canada (email: dzheng@mcrlab.uottawa.ca).

Jiyong Zhao is with Multimedia Communications Lab, School of Information Technology and Engineering, University of Ottawa, 800 King Edward Ave, Ottawa, Ontario, K1N 6N5, Canada (email: jyzhao@site.uottawa.ca).

Abdulmotaleb El Saddik is with Multimedia Communications Lab, School of Information Technology and Engineering, University of Ottawa, 800 King Edward Ave, Ottawa, Ontario, K1N 6N5, Canada (email: abed@mcrlab.uottawa.ca).

very good review on detecting watermarks after geometric transformations.

O’Ruanaidh et al. [9] first have outlined the theory of integral transform invariants and showed that this can be used to produce watermarks that are resistant to rotation, scaling, and translation. In their approach the discrete Fourier transform (DFT) of an image is computed and then the Fourier-Mellin transform is performed on the magnitude, the watermark is embedded in the magnitude of the resulting transform. The watermarked image is reconstructed by performing the inverse transforms (an inverse DFT and an inverse Fourier-Mellin transform) after considering the original phase [7][9]. Fourier-Mellin transform is a log-polar mapping (LPM) followed by a Fourier transform, while an inverse Fourier-Mellin transform is an inverse log-polar mapping (ILPM) followed by an inverse Fourier transform. In the scheme, the embedded watermark may be extracted by transforming the watermarked image into RST invariant domain. However, they noted very severe implementation difficulties which might have hampered further work in this area [8]. According to our experiment, the resulting watermarked images show great ringing effect caused mainly by LPM and ILPM, the quality is definitely unacceptable. O’Ruanaidh et al. noticed this and suggested to embed watermark in the RST invariant domain independently of the original image, so that the original image will not suffer from LPM and ILPM transform theoretically. However it is difficult to implement according to our experiments.

Another strategy for detecting watermarks after geometric distortion is to identify what the distortion were, and invert them before applying the watermark detector. This can be done by embedding a template along with the watermark [8]. Pereira et al. [6] proposed to embed two watermarks, a template and a spread spectrum message containing the information or payload. The template contains no information itself, but is used to detect transformations undergone by the image. “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 quality. A second problem arises because all image watermarked with this method will share a common registration watermark. This fact may improve collusion attempts to discern the registration pattern and, once found, the registration pattern could be removed from all watermarked images thus restricting the invertibility of any geometric distortions.” [8]

Lin et al. [8] proposed a method that develops a wa-

termark invariant to geometric distortions, and that eliminates the need to identify and invert them. The watermark is embedded into a translation and scaling invariant one-dimensional 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. The scheme handles rotations by exhaustive search. We suspect that the probability of false positive of the algorithm is high due to the summing and exhaustive search, and that the exhaustive search is time-consuming. Also rotations of fractional degrees should be considered.

It is widely accepted that phase plays an important, and often crucial, role in vision and image representation [10][11]. Experiments [12] show that images reconstructed from the original phase and the magnitude taken from another different source closely resemble the original ones, unlike the case of images reconstructed from magnitude only. Phase correlation based methods have been proposed to align two images which shifted relative to each other [13][14]. We propose in this paper a scheme that uses phase correlation in LPM domain to rectify the watermark position. The main contributions of the work include the idea of using phase correlation spectrum in digital image watermarking, and the simple and feasible implementation of RST invariant watermarking scheme in LPM domain.

The novel watermarking scheme is invariant to rotation and translation, invariant to scaling when the scale is in a reasonable range, and that is applicable due to its fidelity. In Section II we propose our scheme, in Sections III and IV we describe the watermark embedding and watermark extraction procedures, in Section V we list several strategies used for implementing our scheme, in Section VI we illustrate and evaluate the proposed scheme, and in Section VII we conclude the paper.

II. PROPOSED SCHEME

The DFT (Discrete Fourier Transform) of an image $f(x, y)$ of size $M \times N$ and the corresponding IDFT (Inverse DFT) are defined as follows [15]:

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M+vy/N)} \quad (1)$$

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M+vy/N)} \quad (2)$$

The Fourier spectrum and phase angle are defined as follows:

$$|F(u, v)| = [R^2(x, y) + I^2(x, y)]^{1/2} \quad (3)$$

$$\phi(u, v) = \tan^{-1} \left[\frac{I(u, v)}{R(u, v)} \right] \quad (4)$$

where $R(u, v)$ and $I(u, v)$ are the real and imaginary parts of $F(u, v)$, respectively.

We can write the relationship of an image $i_0(x, y)$, and a rotated, scaled, and translated version of the image, $i_1(x, y)$, as follows [8][9]:

$$\begin{aligned} i_1(x, y) &= i_0(\sigma(x \cos \alpha + y \sin \alpha) - x_0, \\ &\quad \sigma(-x \sin \alpha + y \cos \alpha) - y_0) \end{aligned} \quad (5)$$

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

The Fourier transform of $i_1(x, y)$ and $i_0(x, y)$ are respectively $I_1(u, v)$ and $I_0(u, v)$, and their magnitudes are related by [8][9]:

$$\begin{aligned} |I_1(u, v)| &= |\sigma|^{-2} |I_0(\sigma^{-1}(u \cos \alpha + v \sin \alpha), \\ &\quad \sigma^{-1}(-u \sin \alpha + v \cos \alpha))| \end{aligned} \quad (6)$$

Equ. (6) is independent of the translational parameters (x_0, y_0) , which is the translation property of the Fourier transform [16].

Rewrite Equ. (6) by using log-polar coordinates:

$$u = e^\rho \cos \theta \quad (7)$$

$$v = e^\rho \sin \theta \quad (8)$$

where $\rho \in \mathfrak{R}^2$ and $0 \leq \theta < 2\pi$. For how to calculate ρ and θ , refer to Section V-A. Then the magnitude of the Fourier spectrum can be written as [8][9]:

$$\begin{aligned} |I_1(u, v)| &= |\sigma|^{-2} |I_0(\sigma^{-1} e^\rho \cos(\theta - \alpha), \\ &\quad \sigma^{-1} e^\rho \sin(\theta - \alpha))| \end{aligned} \quad (9)$$

or

$$|I_1(\rho, \theta)| = |\sigma|^{-2} |I_0(\rho - \log \sigma, \theta - \alpha)| \quad (10)$$

Equ. (10) 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 log-radius ρ axis, that image rotation results in a cyclical shift of α along the angle θ axis, and that image translation has no effects in LPM domain.

According to the translation property of the Fourier transform, the Fourier transforms of I_1 and I_0 is related by

$$F_1(\omega_\rho, \omega_\theta) = |\sigma|^{-2} e^{-j(\omega_\rho \cdot \log \sigma + \omega_\theta \cdot \alpha)} F_0(\omega_\rho, \omega_\theta) \quad (11)$$

The Fourier magnitude of the two LPM mappings is related by

$$|F_1(\omega_\rho, \omega_\theta)| = |\sigma|^{-2} |F_0(\omega_\rho, \omega_\theta)| \quad (12)$$

where F_1 and F_0 are respectively the DFT of I_1 and I_0 .

The phase difference between the two LPM mappings is directly related to their displacement, given by $e^{j(\omega_\rho \cdot \log \sigma + \omega_\theta \cdot \alpha)}$.

Equ. (12) is equivalent to computing the Fourier-Mellin transform [9]. Equ. (12) demonstrates that the amplitude of Fourier-Mellin spectrum is scaled by $|\sigma|^{-2}$ caused by scaling transform, and is invariant to rotation and translation. $|\sigma|^{-2}$ will cause no problem at all since we use normalized correlation to detect watermarks, so Fourier-Mellin transform is truly invariant to RST.

The original Fourier-Mellin based watermarking algorithm was proposed by O’Ruanaidh et al. [9]. One of the significant contribution of the paper is the novel application of Fourier-Mellin transform to digital image watermarking. Theoretically Fourier-Mellin domain is the best place to embed watermark, considering it is invariant to RST. In practice, the original image will need to endure both the LPM and ILPM, which make the image quality unacceptable. Ruanaidh et al. noticed the problem, so they proposed an alternative algorithm. Using this algorithm, only the watermark data goes through the ILPM, then it is inserted into the magnitude spectrum of the image. Applying the inverse DFT to the modified magnitude spectrum, one can get the watermarked image. For details of both the algorithms, refer to [9]. However there are several disadvantages in this algorithm, first the watermark data need go through the ILPM which will cause the distortion of the watermark signal. During the embedding process, it is almost impossible to find a good method to insert the watermark data (after the IDFT and ILPM) into the desired location in the image magnitude spectrum. So it is extremely hard to achieve the tradeoff between the invisibility of watermark and the robustness of the watermark.

Based on the second proposal by O’Ruanaidh et al., we propose a new scheme here, as shown in Fig. 1 and Fig. 2. While Section III and Section IV will explain the watermark embedding and extraction scheme in detail, the following outlines several important points of the scheme:

We embed watermark in LPM domain to simplify the effects of RST transformations into simple shifts (refer to Equ. (10)). For watermark embedding, the approximate ILPM is employed to replace ILPM, in order to eliminate the imprecision caused by ILPM. Therefore actually watermarks are embedded in the Fourier magnitude spectrum of the original image, to achieve the effect of being embedded in LPM domain.

Since we do not apply the IDFT before the approximate ILPM, rotation and scaling operation in the spatial domain of the watermarked image will cause translation of the watermark positions in LPM domain, either a circular shift along the angle axis or the vertical shift along the log-radius axis (refer to Equ. (10)). Exhaustive search in the embedding area can be used to handle the shift of watermark positions caused by rotation and scaling [17]. However exhaustive search is time consuming and produces large correlation coefficient for unwatermarked images. Therefore, we use phase correlation to rectify the watermark position

to avoid exhaustive search [18].

Refer to Equ. (10) and Equ. (11), if we compute the cross-power spectrum of F_1 and F_0 as follows [13]:

$$C_{10} = \frac{F_1(\omega_\rho, \omega_\theta) F_0^*(\omega_\rho, \omega_\theta)}{|F_1(\omega_\rho, \omega_\theta) F_0^*(\omega_\rho, \omega_\theta)|} = e^{j(\omega_\rho \cdot \log \sigma + \omega_\theta \cdot \alpha)} \quad (13)$$

where F^* is the complex conjugate of F , the translation property guarantees that the phase of the cross-power spectrum is equivalent to the phase difference between the images. Furthermore, if we represent the phase of the cross-power spectrum in its spatial form, i.e., by taking the inverse Fourier transform of the representation in the frequency domain,

$$D_{10} = IDFT(\text{angle}(C_{10})) \quad (14)$$

where $IDFT$ is inverse Fourier transform, and $\text{angle}(C_{10})$ is the phase of C_{10} .

Based on the property of the Fourier transform, the Fourier transform of function $\delta(x - d)$ is $e^{-j\omega d}$. Equ. (14) gives a two-dimensional δ function centered at the displacement. So D_{10} is a function which is an impulse, that is, it is approximately zero everywhere except at the displacement.

Our method determines the location of peak of D_{10} , and consequently calculate the watermarking position. Since the phase difference for every frequency component contributes equally, the location of the peak will not change if there are noises caused by watermark embedding, black pixel padding, and JPEG compression. The idea eliminates the need for exhaustive search, reduces the correlation coefficient calculated for unwatermarked images, and saves computation time.

III. WATERMARK EMBEDDING

The procedure of embedding a watermark consists of the following steps (refer to Fig. 1):

1. First, use PN generator to generate a watermark data sequence, which is spread spectrum consisting of both positive and negative values.
2. Compute DFT of the original image. The magnitude spectrum of DFT is positive, while the watermark is a sequence of numbers that can be positive or negative. To be able to embed both positive and negative numbers, we need two numbers to represent one original watermark number. We encode positive numbers x as $(x, 0)$ and negative numbers x as $(0, x)$, so the length of watermark data sequence is doubled.
3. Select the desired locations in the LPM magnitude spectrum for embedding the watermark data sequence.
4. If we embed watermark in LPM domain, we need ILPM to transform back from LPM domain to DFT domain. To avoid computing ILPM that may bring unacceptable computational imprecision, we use the approximative ILPM and embed watermark in DFT domain. The watermarking locations in the Cartesian DFT magnitude spectrum is approximated from the watermark points in LPM domain selected in step 3.

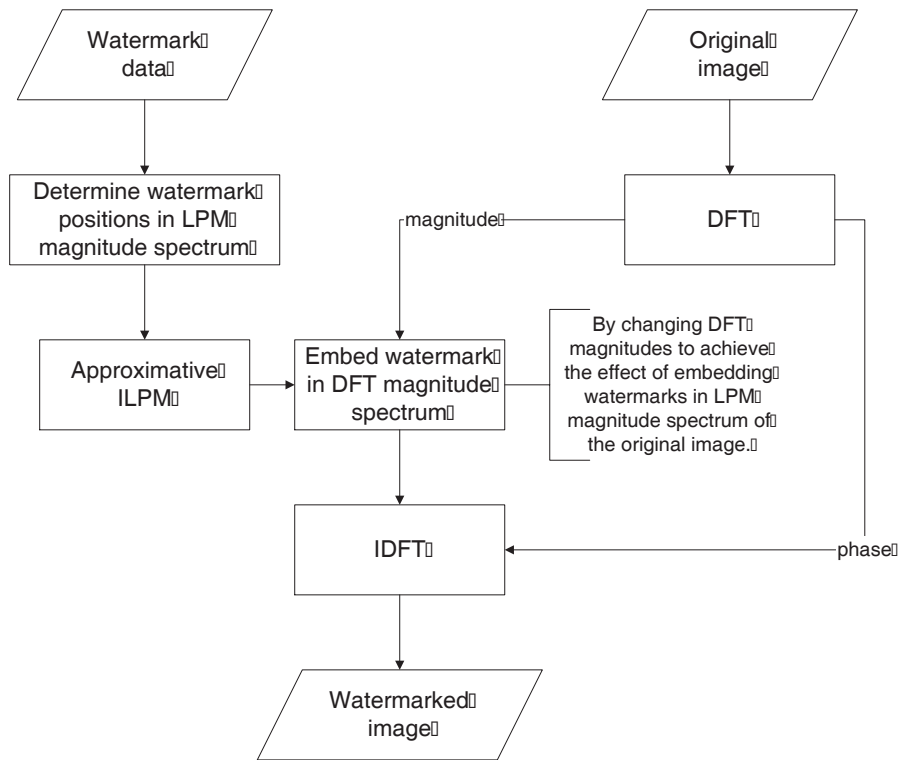


Fig. 1. Watermark embedding scheme.

Naturally, if we want to change the value of one point in LPM magnitude spectrum for embedding watermark data, we only need to find the corresponding four points in Cartesian magnitude spectrum and change their values accordingly. For details, refer to Section V-A.

5. Embed the watermark data into the DFT magnitude spectrum of the original image. Most algorithms use a simple embedding equation such as:

$$E' = E + \alpha * W \quad (15)$$

where E is the DFT magnitude spectrum of the original image, W is the watermark data, E' is the modified DFT magnitude spectrum of the original image, and α is watermarking strength used to achieve the tradeoff between the robustness and the visibility of the watermark.

According to our experiments, by carefully selecting the watermarking positions and watermarking factor β , the difference between E and $\beta * W$ can be small enough. So we can use Equ. (16) to replace the values of the embedding points by $\beta * W$.

$$E' = \beta * W \quad (16)$$

This embedding process will not change the amplitude values of those embedding points dramatically, therefore the goal of invisibility can be achieved. Meanwhile, the embedding method can simplify the extraction process.

The scaling operation will change the value of DFT magnitude spectrum, which is proportional to the scaling factor. The correlation function can be normalized for amplitude changes by using the correlation coefficient, which is in the

range of -1 to 1 , independent of scale changes in the amplitude (refer to Equ. (18)).

6. Finally apply inverse DFT to get the watermarked image.

During the embedding process, the symmetry of the DFT magnitude spectrum should be maintained. For details, refer to Section V-C.

IV. WATERMARK EXTRACTION

Watermark extraction can be done with or without using original image. Exhaustive search can be used if original image is not available [17]. The rotation and scaling transformation in spatial domain result in a cyclically translational shift in LPM domain. So the watermark in LPM domain can only be shifted from its original place. We can use the exhaustive search method to simply shift the whole image in LPM domain vertically and horizontally pixel by pixel, retrieve the “watermark data” at the embedding position, compute the correlation, and choose the largest value as the extraction result.

This section however gives the procedure of watermark extraction by using original image, which outperforms the exhaustive search scheme (refer to Fig. 2):

1. Apply DFT and LPM to both the original and the watermarked image, transform them into LPM domain.
2. Calculate displacement between the LPM of the original image and the LPM of the watermarked image, according to Equ. (13) and Equ. (14). For details, refer to Section V-E.
3. Rectify the original watermark position according to calculated displacement.

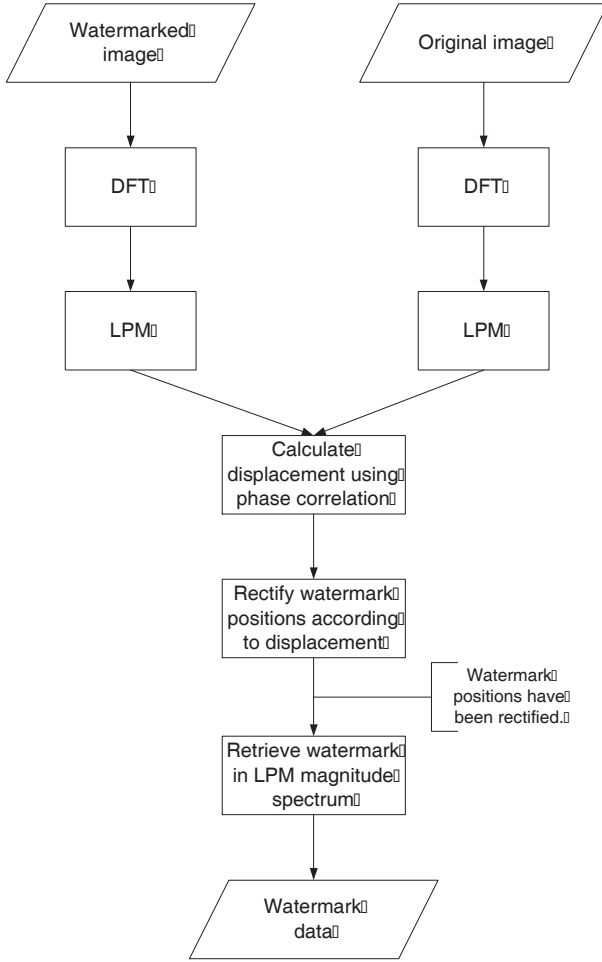


Fig. 2. Watermark extraction scheme.

4. Retrieve the watermark data V at the rectified location by using the following equation:

$$V = \frac{E''}{\beta} \quad (17)$$

where E'' is the DFT magnitude spectrum of the watermarked image that undergone RST transformations and other attacks, while β is defined in Equ. (16). The value change of V caused by a scaling transformation is proportional to the scale factor, and the normalized correlation calculated by Equ. (18) is independent of the change.

5. By using Equ. (18), calculate the normalized correlation coefficient between the original watermark data and the extracted watermark data. If the value of similarity is larger than the threshold, the watermark is successfully extracted, otherwise, the watermark does not exist or we fail to detect it.

$$sim = \frac{W \times V^T}{\sqrt{(W \times W^T)(V \times V^T)}} \quad (18)$$

where W and V are respectively the original watermark vector and retrieved watermark vector, and $(\cdot)^T$ is the transpose operation of a matrix.

V. IMPLEMENTATION STRATEGIES

In this section, we list several of the important problems we met when implementing our scheme proposed in Section II, and our solutions to those problems.

A. LPM and ILPM

In log-polar mapping, pixels are indexed by ring number R and wedge number W , related to ordinary x, y image coordinates by the mapping [19]:

$$r = [(x - x_c)^2 + (y - y_c)^2]^{1/2} \quad (19)$$

$$\theta = \tan^{-1} \frac{y - y_c}{x - x_c}$$

$$R = \frac{(n_r - 1) \log(r/r_{min})}{\log(r_{max}/r_{min})} \quad (20)$$

$$W = \frac{n_w \theta}{2\pi}$$

where (r, θ) are polar coordinates, (x_c, y_c) is the position of the centre of the log-polar sampling pattern, n_r and n_w are the numbers of rings and wedges respectively, and r_{min} and r_{max} are the radii of the smallest and largest rings of samples. We define log-polar radius ρ as:

$$\rho = \ln r. \quad (21)$$

A log-polar sampled image is the one whose samples are centered on points mapping to integral R and W , $R \in \{0, \dots, n_r - 1\}$, $W \in \{0, \dots, n_w - 1\}$. The separation between sample points is proportional to the distance from the sampling center.

Log-polar sampled images are often displayed on orthogonal (R, W) axes, which is also called (ρ, θ) axes in this paper.

The LPM can be explained by the following equation:

$$P = L * C \quad (22)$$

where P and C are respectively the points in LPM magnitude spectrum and the points in Cartesian magnitude spectrum, while L is the LPM computation operator.

In this scheme, the LPM and ILPM are the major causes of the image quality loss. Using bilinear interpolation, each point in log-polar magnitude spectrum is computed from a weighted average of four points in the Cartesian magnitude spectrum, shown in Equ. (23) and Fig. 3.

$$\begin{aligned} P(\rho, \theta) &= C(x, y) \cdot (1 - a) \cdot (1 - b) \\ &+ C(x, y + 1) \cdot (1 - a) \cdot b \\ &+ C(x + 1, y) \cdot a \cdot (1 - b) \\ &+ C(x + 1, y + 1) \cdot a \cdot b \end{aligned} \quad (23)$$

where $C(x, y)$, $C(x, y + 1)$, $C(x + 1, y)$, and $C(x + 1, y + 1)$ are four points in Cartesian coordinate, $P(\rho, \theta)$ (P in Fig. 3) is the corresponding point inside the square specified

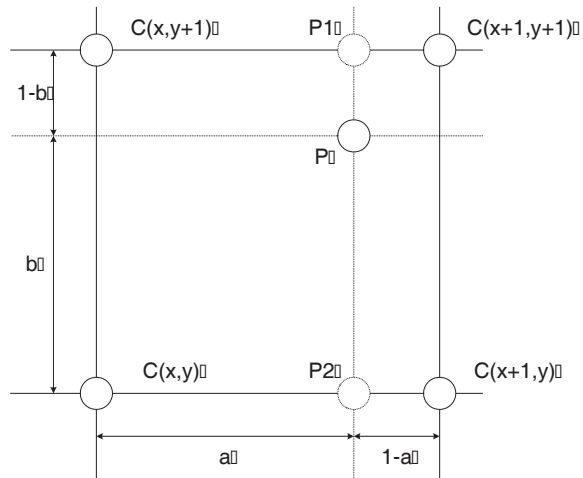


Fig. 3. Bilinear interpolation.



(a) LPM domain (b) Cartesian domain

Fig. 4. Proposed watermark embedding position.

by the four points, and a and b are respectively the x-axis and y-axis coordinate difference between point P and point $C(x, y)$.

If watermark data is embedded in log-polar magnitude spectrum, we need ILPM to get the corresponding position array in the Cartesian magnitude spectrum. We use an approximate ILPM instead to avoid computational imprecision.

Suppose that we want to insert watermark M at position $P(\rho, \theta)$, so the value of $P(\rho, \theta)$ should be added by M . From the value of ρ and θ , we can get the exact corresponding value of x and y , then add M to each of the four points $C(x, y)$, $C(x, y + 1)$, $C(x + 1, y)$, and $C(x + 1, y + 1)$ in the Cartesian magnitude spectrum. Thus we can be sure that after log-polar mapping, the value of $P(\rho, \theta)$ is added exactly by M .

B. Watermark positions

Frequency domain watermarking is useful for taking advantage of perceptual criteria in the embedding process, for designing watermarking techniques which are robust to common compression techniques, and for direct watermark embedding of compressed bit streams [20]. For the digital watermarking scheme in frequency domain, to make the watermark robust to the attacks such as lossy compression

and filter processing which may remove the high-frequency components in the magnitude spectrum, Cox et al. [21] proposed that the watermark should be embedded in the most significant frequency components. Thus the watermark can resist most attacks of lossy compression and filter processing. However, to make the watermark invisible and keep the fidelity of the image, we need to embed the watermark into the least significant frequency components. To get the tradeoff, we choose the middle frequency components as the location to insert watermark data.

Meanwhile, because the log-polar mapping is just like a sampling process, the closer to the center, the higher the sampling rates. So if we insert the watermark data into the low frequency components, the change of the value of one point in the Cartesian magnitude spectrum will cause value changes of a lot of points in the log-polar magnitude spectrum because of the bilinear interpolation. That may cause imprecision in the extraction process.

So in our watermarking scheme, we use a simple and effective empirical perceptual model to embed watermark data into the middle frequency components. Experiments show the effectiveness of this approach.

For security issues, we can randomly choose to insert the watermark data into the points in the white region of LPM magnitude spectrum, as shown in Fig. 4 (a). Accordingly, the positions of points in Cartesian magnitude spectrum can be determined. Those points are located in the white region of Cartesian magnitude spectrum, as shown in Fig. 4 (b). The points are located in the middle frequency range, the tradeoff between the robustness and image fidelity can be achieved. The locations of these points can be randomly determined by a security key, without which the exact location of these points cannot be known. These points are randomly located in a region, therefore it is hard to retrieve the position information through observation, and it is hard to remove or attack the watermark maliciously.

The number of the watermark points depends on the length of the PN sequence, which is 64 in our experiments. Refer to Section III, the length of watermark sequence is doubled in order to be able to embed both positive and negative watermark data. Refer to Section III and Section V-A, we embed at four points in DFT domain to achieve the effect of embedding at one point in LPM domain. Refer to Section V-C, the number of watermark points is doubled in order to maintain the frequency symmetry. Therefore, 1024 watermark points are used for embedding watermark in our experiments.

C. Symmetry of watermark embedding

We should maintain the frequency symmetry when we embed watermark in the Fourier frequency domain. Refer to Section III and Section V-A, we embed at four points in DFT domain to achieve the effect of embedding at one point in LPM domain. Refer to Fig. 5, if we want to embed four points d_{11} , d_{12} , d_{21} , and d_{22} in DFT domain, we need embed in the same strength at points d'_{11} , d'_{12} , d'_{21} , and d'_{22} . Here d'_{11} , d'_{12} , d'_{21} , and d'_{22} are respectively symmetric about the origin with d_{11} , d_{12} , d_{21} , and d_{22} . Doubling the

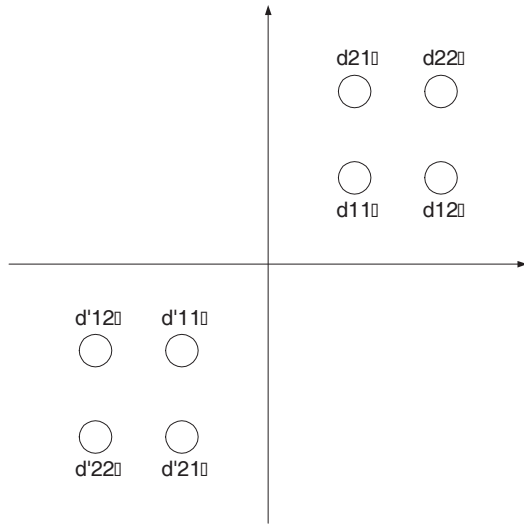


Fig. 5. Symmetry of watermark embedding.

watermarking points will not necessarily reduce the image quality, because we can use weaker embedding strength and take advantage of the data redundancy. Keeping symmetry enhances the performance of watermark extraction on rotation transformations.

D. Removal of low frequency components

From the watermark embedding process, we clearly know where the watermark data could be, so after the first DFT transform, we can discard the central part of the DFT amplitude spectrum. Such a procedure will not lose the watermark data and has advantages according to our experiments.

The amplitudes of the low frequency components are much larger than the high and middle frequency components. After the scaling and rotation operations, the changes in the low frequency components are overwhelming in the entire changes of the amplitude spectrum. And if there are transformations such as cropping and new boundary, which are usual when the scaling and rotation are applied to the image, such undesired changes in the low frequency components maybe even more obvious according to our observations. Removal of low frequency components helps to produce more accurate displacement calculation.

E. Displacement calculation and watermarking position rectification

Refer to Equ. (13) and Equ. (14), we calculate C_{10} then D_{10} . The LPM of the watermarked image undergone rotational and/or scaling transformation and the LPM of the original image are not simply translated (displaced) from each other, actually the former is a cyclically shifted version of the latter. Therefore, two peaks in the phase correlation spectrum are expected (refer to Fig. 6 and Fig. 9). The algorithm does not discriminate one peak from another, instead both peaks are declared as displacement. Fig. 6 illustrates the peaks in resulting phase correlation spectrum after the watermarked image has been undergone rotation,

scaling, and rotation and scaling transformations. Theoretically the two peaks in Fig. 6 (a) are symmetric about the vertical line passing through the image center, the two peaks in Fig. 6 (b) are symmetric about the horizontal line passing through the image center, the two peaks in Fig. 6 (c) are symmetric about the image center. We can use this feature to rectify inaccuracy in the displacement calculation if any. It is very easy to detect the peaks if there is some displacement, since the peaks usually have very big value comparing the rest. If the watermarked image has not undergone rotation or scaling transformation, there is no displacement. In this situation, all the elements of array D_{10} will be approaching zero, so that we cannot detect peaks in array D_{10} . However, when two images match the elements of array C_{10} (cross-power spectrum) should be all approaching 1. By using this feature, we can judge that there is no displacement. We declare both the detected peaks and all their eight neighbors as displacement, to make our algorithms tolerate peak-detection inaccuracy. Using those eight neighbors as displacement could generate a slightly bigger normalized correlation for unwatermarked images, but the effect is neglectable. Experiments have demonstrated that the proposed method is very reliable.

F. Threshold selection

Since the normalized correlation is used as the detection measure, two methods can be used to estimate the false positive probability [22]. According to those methods, the threshold T can be set by first determining what false positive probability P_{fpp} is required in application.

The first is the approximate Gaussian method. Assuming the watermark sequence is zero mean, uncorrelated with the original image and n , the length of the watermark sequence, is large enough, then under hypothesis that the watermark does not exist, the normalized correlations can be approximated as a Gaussian distribution with standard deviation $1/\sqrt{n}$ and zero mean based on the central limit theorem.

$$P_{fpp} = \frac{1}{\sqrt{2\pi}\sigma_0} \int_T^{\infty} e^{-\frac{(x-m_0)^2}{2\sigma_0^2}} dx = Q\left(\frac{T-m_0}{\sigma_0}\right) \quad (24)$$

where mean $m_0 \approx 0$, standard deviation $\sigma_0 = 1/\sqrt{n}$, T is the threshold and Q is defined as:

$$Q(X) = \frac{1}{\sqrt{2\pi}} \int_X^{\infty} e^{-\frac{x^2}{2}} dx \quad (25)$$

However, as the threshold increases, the approximate Gaussian method begins to dramatically overestimate the false positive probability [22].

We use the method proposed by Miller et al. [22] to give a more accurate estimation about the false positive probability. The method is characterized by the following equations:

$$P_{fpp} = \frac{I_{n-2}(T_a)}{2I_{n-2}(\frac{\pi}{2})} \quad (26)$$

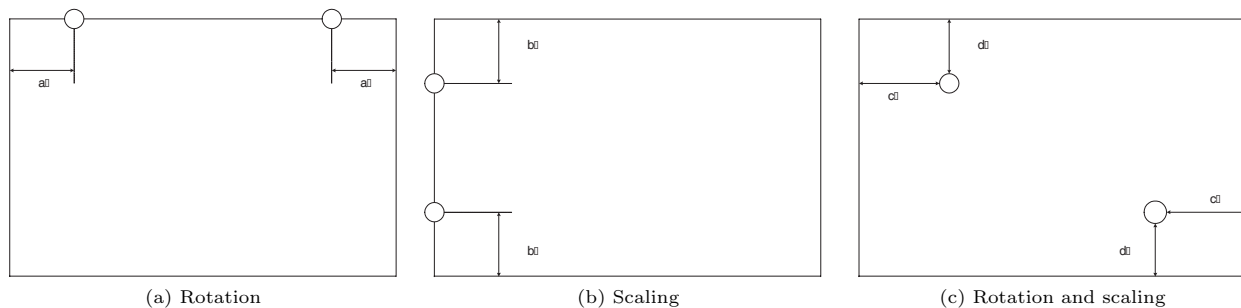


Fig. 6. Phase correlation illustration.

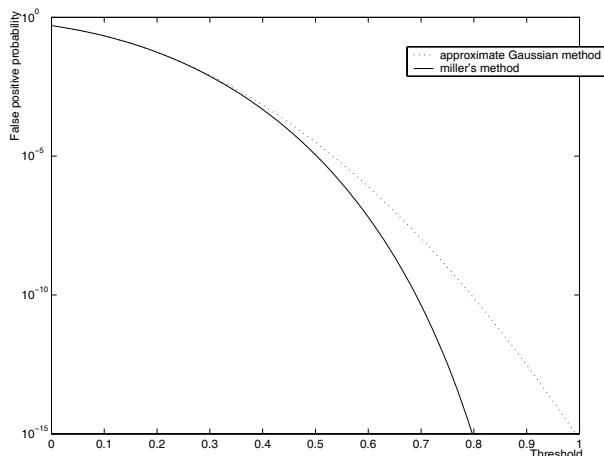


Fig. 7. False positive probability versus threshold T . The length of watermark sequence is 64. The solid curve represents the Miller's method, while the dotted curve represents the approximate Gaussian method.

$$I_d(\theta) = \int_0^\theta \sin^d(u) du \quad (27)$$

$$T_a = \cos^{-1}(T) \quad (28)$$

The relationship between T and P_{fpp} described in the above equations is illustrated in Fig. 7.

Based on the Miller's method, when the thresholds are 0.4, 0.5, and 0.6, the corresponding false positive probabilities are respectively 4.8×10^{-4} , 1.1×10^{-5} , and 6.4×10^{-8} . The lower the threshold, the more likely the false positive error will happen. Conversely, the higher the threshold, the more likely the false negative error will happen. In our experiments, we set the threshold to 0.5, by which the false positive probability smaller than 1.1×10^{-5} can be achieved.

VI. EXPERIMENTAL RESULTS AND EVALUATIONS

In this section, we illustrate and evaluate the performance of the proposed scheme against rotation, scaling, and translation transformations, and robustness against JPEG compression and other attacks.

A. The original image and watermarked image

We use image *Barbara* as the original test image, shown as Fig. 8 (a). The watermarked image is shown as Fig. 8 (b), which is obtained by embedding using such a strength that the watermark is just imperceptible. The PSNR of Fig. 8 (b) is 44.2070dB. We experimented with various β values, and we found that β values in the range of 150 to 200 give the best objective and subjective qualities to the test images. In our experiments, we use $\beta = 150$. All the following experiments will be conducted on this watermarked image. We also have done experiments on other test images, and the similar results have been achieved. Test results will show that the scheme can meet the requirements of both imperceptibility and robustness.

B. Displacement calculation: rotation

Fig. 9 (a) is transformed from the watermarked image (Fig. 8 (b)), by rotating the watermarked image by 45° counter-clockwise without scaling. Fig. 9 (b) is the amplitude of the corresponding phase correlation spectrum. Black pixels have been padded to the outside of the three transformed images to maintain the shape and size of the entire resulting image.

For comparison purpose, the displacement calculated by using Equ. (19) and Equ. (20) is (1, 65), where 1 and 65 are respectively the row number and column number of the displacement. Note the indexes start from 1. (1, 65) here means that the log-polar radius ρ in LPM of the watermarked image has not been changed, while the angle θ in LPM of the watermarked image has been cyclically shifted 65 columns right (rotated 45° counter-clockwise).

We applied DFT and LPM to both the original image (Fig. 8(a)) and the rotated watermarked image (Fig. 9 (a)), then obtained the phase correlation spectrum between the LPM of the original and that of the watermarked, shown as Fig. 9 (b). There are two peaks in the phase correlation spectrum. The phase correlation of the first peak at (1, 65) is $+0.0000 + 0.1166i$, and the phase correlation of the second peak at (1, 449) is $+0.0000 - 0.1166i$. We use these two displacements to rectify the watermarking positions.

After the rectification, we calculate and get the correlation coefficient of 0.7078, which is the same as the result of the exhaustive search.



(a) Barbara: the original test image

(b) The watermarked image, $PSNR = 44.2070dB$. All the following tests were based on this image.

Fig. 8. The original image and watermarked image.



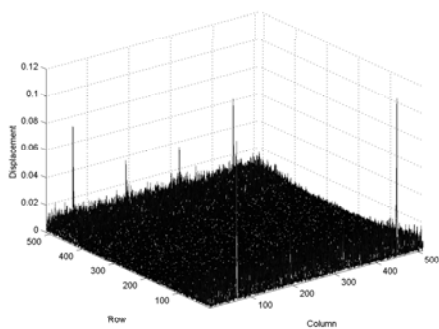
(a) Rotated by 45 degree counter-clockwise without scaling. Four corners of the image have been cropped.



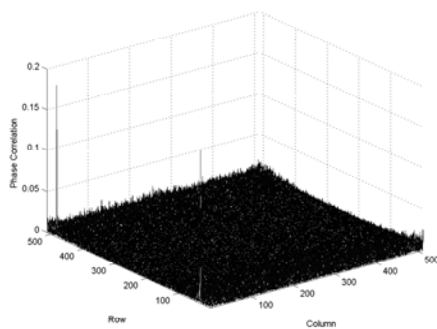
(c) Scaled by 0.7.



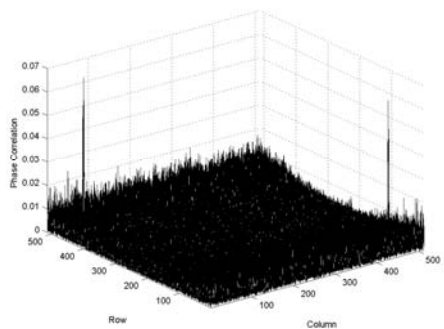
(e) Rotated by 45 degree counter-clockwise after being scaled by 0.7062.



(b) Phase correlation between LPM of (a) and LPM of the original image. There are two peaks, one at (1, 65) and another at (1, 449).



(d) Phase correlation between LPM of (c) and LPM of the original image. There are two peaks, one at (32, 1) and another at (482, 1).



(f) Phase correlation between LPM of (e) and LPM of the original image. There are two peaks, one at (482, 65) and another at (32, 449).

Fig. 9. Phase correlation for watermarked images undergone different transformations.

C. Displacement calculation: scaling without rotation

Fig. 9 (c) is transformed from the watermarked image (Fig. 8 (b)), by scaling the watermarked image by 0.7. Fig. 9 (d) is the amplitude of the corresponding phase correlation spectrum.

For comparison purpose, the displacement calculated by using Equ. (19) and Equ. (20) is (482, 1), which means that the log-polar radius ρ in LPM of the watermarked image has been cyclically shifted 482 columns down (scaled by 0.7), while the angle θ in LPM of the watermarked image has not been changed.

We applied DFT and LPM to both the original image (Fig. 8(a)) and the rotated watermarked image (Fig. 9 (c)), then obtained the phase correlation spectrum between the LPM of the original and that of the watermarked, shown as Fig. 9 (d). There are two peaks in the phase correlation spectrum. The phase correlation of the first peak at (32, 1) is $-0.0000 - 0.1863i$, and the phase correlation of the second peak at (482, 1) is $-0.0000 + 0.1863i$. We use these two displacements to rectify the watermarking positions.

After the rectification, we calculate and get the correlation coefficient of 0.8335, which is the same as the result of the exhaustive search.

D. Displacement calculation: scaling and rotation

Fig. 9 (e) is transformed from the watermarked image (Fig. 8 (b)), by rotating the watermarked image by 45° counter-clockwise after scaling by 0.7062. Fig. 9 (f) is the amplitude of the corresponding phase correlation spectrum.

For comparison purpose, the displacement calculated by using Equ. (19) and Equ. (20) is (482, 65), which means that the log-polar radius ρ in LPM of the watermarked image has been cyclically shifted 482 columns down (scaled by 0.7), while the angle θ in LPM of the watermarked image has been cyclically shifted 65 columns right (rotated 45° counter-clockwise).

We applied DFT and LPM to both the original image (Fig. 8(a)) and the rotated watermarked image (Fig. 9 (e)), then obtained the phase correlation spectrum between the LPM of the original and that of the watermarked, shown as Fig. 9 (f). There are two peaks in the phase correlation spectrum. The phase correlation of the first peak at (482, 65) is $-0.0000 - 0.0648i$, and the phase correlation of the second peak at (32, 449) is $-0.0000 + 0.0648i$. We use these two displacements to rectify the watermarking positions.

After the rectification, we calculate and get the correlation coefficient of 0.8119, which is the same as the result of the exhaustive search.

E. Experimental results: rotation with cropping

Refer to Fig. 9 (a) for the illustration of the experiment. We rotated the watermarked image (Fig. 8 (b)) counter-clockwise by different angles listed in Table I. The four corners of the watermarked image have been cropped off due to the rotation, and black pixels have been padded in

TABLE I
ROTATION WITH CROPPING

Angle	Correlation ₁	Correlation ₂
0°	0.9860	0.1952
0.5°	0.8219	0.1976
1°	0.8613	0.2360
1.5°	0.8251	0.2928
2°	0.8413	0.2236
5°	0.9288	0.2223
5.5°	0.8269	0.2782
10°	0.8508	0.2569
10.5°	0.8476	0.2624
20°	0.8407	0.2551
30°	0.7873	0.3145
40°	0.7627	0.2223
45°	0.7078	0.2697
50°	0.7555	0.1715
60°	0.6451	0.1260
70°	0.6749	0.1599
80°	0.7103	0.2969
90°	0.8626	0.2919
90.5°	0.6926	0.2141
100°	0.7604	0.2921
110°	0.7127	0.2009
120°	0.7416	0.2980
130°	0.7027	0.2932
140°	0.7527	0.3135
150°	0.7349	0.1914
160°	0.8724	0.2386
170°	0.8981	0.1975
180°	0.9860	0.1952
180.5°	0.8219	0.1976

order to maintain the image size and shape.

In all the tables of this section, $Correlation_1$ is the normalized correlation coefficient between the original watermark sequence and the watermark sequence detected from the watermarked image, and $Correlation_2$ is the normalized correlation coefficient between the original watermark sequence and ‘the watermark sequence’ detected from the corresponding unwatermarked image that undergone the same transformations as the watermarked image.

In Table I, the *Angle* is the angle we rotated the watermarked image before extracting watermark, We only listed the results for rotation angles up to 180.5° , since

$$Correlation(180^\circ + \alpha) = Correlation(\alpha). \quad (29)$$

From the table, we can see that the correlation coefficients for the watermarked image are all greater than 0.6451, and the correlation coefficients for the unwatermarked image are all smaller than 0.3145. The algorithm can handle fractional degree rotation. When rotational degree becomes bigger, larger areas need to be cropped and more black pixels need to be padded, therefore $Correlation_1$ tends to be smaller. The $Correlation_2$ values are random noise under about 0.3. We can see from Table I, that the majority of the $Correlation_1$ values are bigger than 0.7, and the majority of the $Correlation_2$ values are smaller than 0.3.

We have done the similar tests to other standard test images, and we obtained the similar results. Therefore we can conclude that the scheme is robust to rotation.

F. Experimental results: scaling

TABLE II
SCALING

Scale	Correlation ₁	Correlation ₂
0.6	0.7834	0.1596
0.7	0.8335	0.1779
0.8	0.8469	0.2752
0.9	0.8839	0.1913
1.0	0.9860	0.1952
1.1	0.9013	0.2873
1.2	0.8401	0.1704
1.3	0.8511	0.2882

Refer to Fig. 9 (c) for illustration. We scaled the watermarked image (Fig. 8 (b)) by using the scales listed in the column *Scale* of the Table II. From the table, we can see that all the *Correlation₁* are greater than 0.7834, and all the *Correlation₂* are less than 0.2882. Therefore we can conclude that the scheme is robust to scaling when the scale factor is in a reasonable range.

G. Experimental results: scaling and rotation

TABLE III
SCALING AND ROTATION

Angle/Scale	Correlation ₁	Correlation ₂
0°/1.0000	0.9860	0.1952
0.5°/0.9903	0.9148	0.2908
1°/0.9827	0.7956	0.2701
1.5°/0.9679	0.8054	0.1735
2°/0.9609	0.8808	0.2895
5°/0.9192	0.8332	0.3258
5.5°/0.9127	0.8268	0.2758
10°/0.8634	0.8541	0.2478
10.5°/0.8576	0.8185	0.2730
20°/0.7793	0.7860	0.2895
30°/0.7304	0.7966	0.2790
40°/0.7107	0.7120	0.1750
45°/0.7062	0.8119	0.2750
50°/0.7107	0.7625	0.2574
60°/0.7304	0.7593	0.2363
70°/0.7793	0.7728	0.2378
80°/0.8634	0.7536	0.2013
90°/1.0000	0.8626	0.3031
90.5°/0.9903	0.8128	0.0956
100°/0.8634	0.7495	0.2278
110°/0.7793	0.7222	0.1862
120°/0.7304	0.7331	0.1725
130°/0.7107	0.7200	0.2357
140°/0.7107	0.8081	0.2948
150°/0.7304	0.7499	0.2224
160°/0.7793	0.7724	0.3099
170°/0.8634	0.8601	0.2444
180°/1.0000	0.9860	0.1952
180.5°/0.9903	0.9148	0.2908

There was no cropping in this test, because the image was shrunk accordingly before being rotated. Refer to Fig. 9 (e) for illustration, black pixels have been padded in order to maintain the image size and shape. The test results are shown in Table III, where the *Angle* is the angle we rotated counterclockwise the watermarked image before extracting watermark, *Scale* is the scaling factor used for scaling before rotation in order to keep the whole image inside the original frame.

From the table, we can see that the correlation coefficients for the watermarked image are all greater than 0.7120, and the correlation coefficients for the unwatermarked image are all smaller than 0.3258. We have done the similar tests to other standard test images, and we obtained the similar results. Therefore we can conclude that the scheme is robust to rotation and scaling.

H. Experimental results: scaling by 0.7 and translation

In this experiment, we shrank the watermarked image by scale 0.7, and then translated it. Black pixels have been padded in order to maintain the image size.

The test results are shown in Table IV, from which we can see that the correlation coefficients for both watermarked image and unwatermarked image are constant. Therefore the scheme is invariant to translation.

TABLE IV
SCALING BY 0.7 AND TRANSLATION

Translation	Correlation ₁	Correlation ₂
no translation	0.8335	0.1779
50 pixels up	0.8335	0.1779
50 pixels down	0.8335	0.1779
50 pixels left	0.8335	0.1779
50 pixels right	0.8335	0.1779
50 pixels right & up	0.8335	0.1779
50 pixels left & up	0.8335	0.1779
50 pixels left & down	0.8335	0.1779
50 pixels right & down	0.8335	0.1779

I. Experimental results: JPEG compression

We compressed the watermarked image *Barbara* (Fig. 8 (b)) by different quality factors, the test results are shown in Table V. From the table, we can see that the results are very good until the quality factor equal to 5%. We think that the robustness to JPEG compression is due to our embedding watermarks in mid-range frequencies.

TABLE V
JPEG COMPRESSION

Quality	Correlation ₁	Correlation ₂
No compression	0.9860	0.1952
90 %	0.9582	0.2216
80 %	0.9823	0.2102
70 %	0.9804	0.2580
60 %	0.9766	0.3218
50 %	0.9681	0.2023
40 %	0.9605	0.2542
30 %	0.9394	0.2586
20 %	0.8579	0.1725
10 %	0.7561	0.2004
5 %	0.3189	0.2360

J. Performance on different images

We applied our watermarking scheme to 100 test images downloaded from the Internet. The image set was chosen to contain a variety of natural images. The images were compressed by JPEG using quality factor equal

to 50%, scaled by 0.7793, and then rotated counterclockwise by 20° . The test results are shown in Fig 10. The horizontal axis shows different images, while the vertical axis shows the corresponding correlation. The upper curve represents the correlation between the original watermark sequence and the watermark sequence detected from the watermarked image, and the lower curve represents the correlation between the original watermark sequence and 'the watermark sequence' detected from the corresponding unwatermarked image. The scheme could still detect correctly from all the watermarked images without false positive from unwatermarked images, under the quite severe condition.

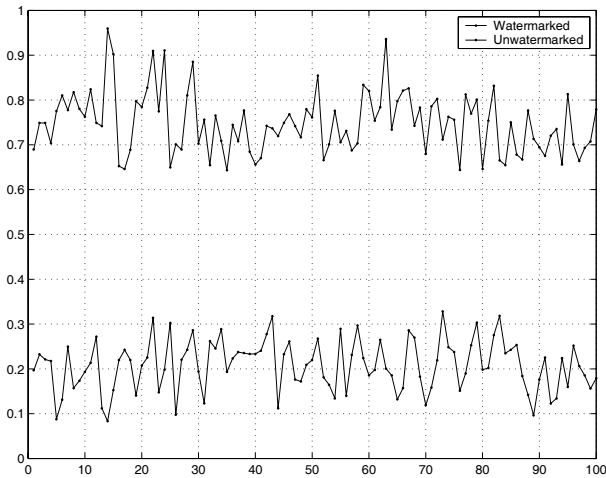


Fig. 10. Watermark detection results for 100 test images.

We have tested our scheme against different transformations and JPEG compression, and the excellent performance of the scheme has been demonstrated.

K. Experiment results: random watermark test

We have conducted random-watermark false positive tests in order to demonstrate the ability of our watermark extraction algorithm, as shown in Fig. 11, where the x-axis represents 1000 randomly generated PN sequences and the y-axis shows the resulting detection values. The test was on Fig. 8 (a), and the embedding strength is the same as the one used to generate Fig. 8 (b). Here, the large difference between the value obtained from the PN sequence originally embedded (shown at location 20 on the x-axis) and the other PN sequences is presented as a demonstration of the ability of the watermarking system to distinguish different PN sequences.

L. Experiment results: performance on different watermarks

We have test our watermarking system on different watermarks in order to demonstrate the good results are not random, as shown in Fig. 12, where the x-axis represents 1000 randomly generated and embedded PN sequences and the y-axis shows the resulting detection values. The test was on Fig. 8 (a), and the embedding strength is the same

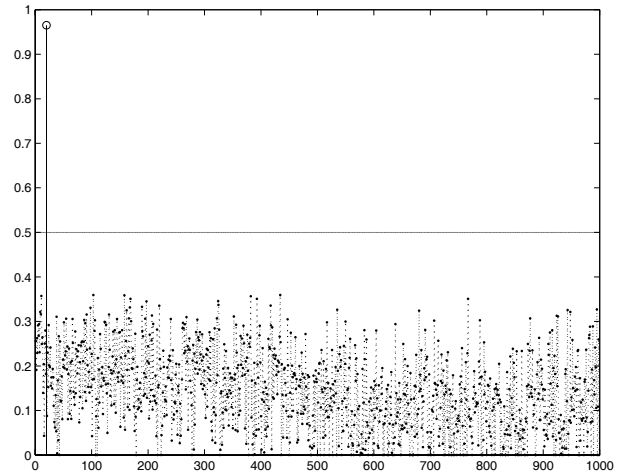


Fig. 11. Watermark detection results for 1000 PN sequence including the one originally embedded.

as the one used to generate Fig. 8 (b). Here, all the detection values are bigger than 0.9. So the watermarking system performs well on any PN sequence.

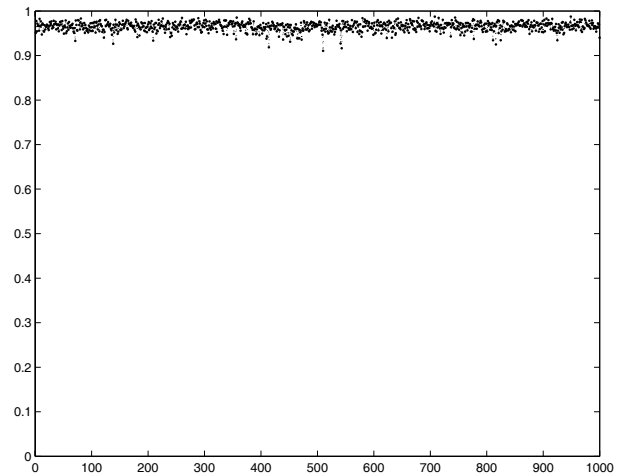


Fig. 12. Watermark detection results for 1000 watermarks.

M. Experiment results: miscellaneous attacks

The watermark scheme was tested with many kinds of attacks including noise pollution, noise removing operation, filter operation, and pixel removing attacks. StirMark was used for generating the attacks. The experiments show the robustness of the proposed watermarking scheme. Due to the embedding mechanism, the robustness of the embedded watermark is not compromised by the ordinary image processing techniques presented in Table VI. For pixel removal in the table, we randomly removed 10 columns or 10 rows for all 5 test cases.

VII. CONCLUSIONS AND FUTURE WORKS

Fourier-Mellin transform is an excellent theory for RST invariant watermarking scheme, however it is difficult to implement due to the impracticality of implementing LPM

TABLE VI
MISCELLANEOUS ATTACKS

<i>Attack</i>	<i>Correlation</i> ₁	<i>Correlation</i> ₂
Gaussian white noise: N(0, 0.001)	0.9399	0.2177
Gaussian white noise: N(0, 0.005)	0.8748	0.2513
Gaussian white noise: N(0, 0.01)	0.6760	0.1512
Gaussian filter: filter size [7 7], standard deviation 0.5	0.9659	0.1963
Gaussian filter: filter size [7 7], standard deviation 1	0.9507	0.1967
Gaussian filter: filter size [3 3], standard deviation 1	0.9605	0.1786
1. Gaussian noise pollution: N(0, 0.001) 2. Wiener Filter to remove noise	0.9101	0.2513
1. Salt & pepper pollution: noise density 0.001 2. Median filter to remove noise	0.8887	0.2352
1. Salt & pepper pollution: noise density 0.01 2. Median filter to remove noise	0.8846	0.2292
1. Salt & pepper pollution: noise density 0.1 2. Median filter to remove noise	0.8005	0.3347
Pixel removal: Case1	0.9580	0.2733
Pixel removal: Case2	0.9540	0.2471
Pixel removal: Case3	0.9647	0.2129
Pixel removal: Case4	0.9676	0.2015
Pixel removal: Case5	0.9661	0.2592

and ILPM. In this paper, we proposed a LPM and phase correlation based digital watermarking scheme that is invariant to RST transformations. We calculate the displacement by detecting peaks in phase correlation spectrum and then rectify the shifts of angle θ and log-radius ρ in LPM magnitude spectrum, and consequently to make the scheme invariant to RST transformations. The test results demonstrated that the scheme is very reliable in displacement calculation and invariant to rotation and translation, invariant to scaling when the scale is in a reasonable range, and very robust to JPEG compression and other attacks. The main contributions of the work include the idea of using phase correlation spectrum in digital image watermarking, and the simple and feasible implementation of RST invariant watermarking scheme in LPM domain.

As for our future work, we will improve our watermark embedding algorithm to give largest possible *Correlation*₁ for all situations, and improve our watermark extraction algorithm to better deal with cropping and black pixel padding caused by rotation and scaling.

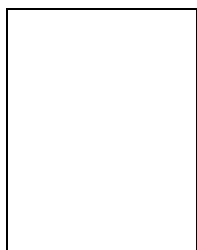
ACKNOWLEDGMENTS

The project was in part supported by Natural Science and Engineering Research Council of Canada. We would like to thank Miss Ronghui Tu for helping with the experiments and data collection.

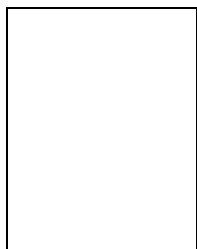
REFERENCES

- [1] I. Cox, J. Kilian, T. Leighton, and T. Shamon, Secure spread spectrum watermarking for multimedia, *IEEE Transactions on Image Processing*, Vol. 6, No. 12, pp. 1673–1687, 1997.
- [2] F. Hartung and B. Girod, Digital watermarking of raw and compressed video, *Proc. European EOS/SPIE Symposium on Advanced Imaging and Network Technologies*, Vol. 2952, pp. 205–213, 1996.
- [3] M. Kutter, Watermarking resistant to translation, rotation, and scaling, *Proceedings of SPIE*, Vol. 3628, pp. 423–431, 1998.
- [4] F. Hartung and M. Kutter, Multimedia watermarking techniques, *Proceedings of the IEEE*, Vol. 87, No. 7, pp. 1079–1107, 1999.
- [5] J. Zhao, R. Hayasaka, R. Muranoi, M. Ito, and Y. Matsushita, A Video Copyright Protection System based on ContentID, *IEICE Transactions on Information and Systems*, Vol.E83-D, No.12, pp. 2131–2141, 2000.
- [6] S. Pereira and T. Pun, Robust Template Matching for Affine Resistant Image Watermarks, *IEEE Transactions on Image Processing*, Vol.9, No.6, pp. 1123–1129, 2000.
- [7] M. Alghoniemy and A. Tewfik, Progressive Quantized Projection Watermarking Scheme, *Proceedings of ACM Multimedia 99*, Vol.1, pp. 295–298, 1999.
- [8] C. Lin, M. Wu, J. Bloom, I. Cox, M. Miller, and Y. Lui, Rotation, Scale, and Translation Resilient Watermarking for Images, *IEEE Transactions on Image Processing*, Vol.10, No.5, pp. 767–782, 2001.
- [9] J. O’Ruanaidh and T. Pun, Rotation, Scale, and Translation Invariant Digital Image Watermarking, *Signal Processing*, Vol.66, No.3, pp. 303–317, 1998.
- [10] J. Behar, M. Porat, and Y. Zeevi, Image reconstruction from localized phase, *IEEE Transactions on Signal Processing*, Vol. 40, No.4, pp. 736–743, 1992.
- [11] S. Urieli, M. Porat, and N. Cohen, Image characteristics and representation by phase: from symmetric to geometric structure, *Proceedings of International Conference on Image Processing 1996*, Vol. 1, pp. 705–708.
- [12] A. Oppenheim and J. Lim, Importance of phase in signals, *Proceedings of IEEE*, Vol. 69, No.5, pp. 529–541, 1981.
- [13] L. G. Brown, A survey of image registration techniques, *ACM Computing Surveys*, Vol. 24, No. 4 pp. 325–376, 1992.
- [14] B. S. Reddy and B. N. Chatterji, A FFT-based technique for translation, rotation, scale-invariant image registration, *IEEE Transactions on Image Processing*, Vol. 5, No. 8 pp. 1266–1271, 1996.
- [15] R. Gonzalez and R. Woods, *Digital Image Processing*, Prentice-Hall, 2002, ISBN: 0-201-18075-8.
- [16] R. Bracewell, *The Fourier Transform and Its Applications*, McGraw-Hill, 2000, ISBN: 0-07-303938-1.
- [17] D. Zheng and J. Zhao, LPM-Based RST Invariant Digital Image Watermarking, *The IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) 2003*, 2003.
- [18] D. Zheng and J. Zhao, RST Invariant Digital Image Watermarking: Importance of Phase Information, *The IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) 2003*, 2003.

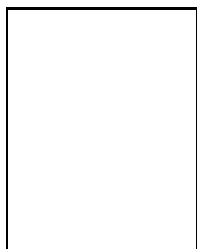
- [19] D. Young, Straight lines and circles in the log-polar image, BMVC2000: Proceedings of the 11th British Machine Vision Conference, pp. 426-435, 2000.
- [20] C. Podilchuk and E. Delp, Digital watermarking: algorithms and applications, IEEE Signal Processing Magazine, Vol., No. pp. 33-46, 2001.
- [21] I. Cox, M. Miller, and J. Bloom, Digital Watermarking, Morgan Kaufmann Publishers, 2002, ISBN: 1-55860-714-5.
- [22] M. Miller and J. Bloom, Computing the probability of false watermark detection, Proceedings of the Third International Workshop on Information Hiding, Dresden, Germany, 1999.
- [23] W. Zeng and B. Liu, A Statistical Watermark Detection Technique without Using Original Images for Resolving Rightful Ownerships of Digital Images, IEEE Transactions on Image Processing, Vol. 8, No.11, pp. 1534-1548, 1999.



Dong Zheng received his B.E., M.E. in Electrical Engineering from Wuhan University, China, respectively in 1996 and 2000. He is now studying for his Master's degree in Computer Engineering in School of Information Technology and Engineering (SITE) at University of Ottawa, Canada. His current research interests are on digital image processing. He is a student member of IEEE.



Jiying Zhao received his B.E., M.E. in Computer Engineering, and Ph.D. degree in Electrical Engineering from North China Electric Power University (NCEPU), China, respectively in 1983, 1988, and 1992, and his second Ph.D. degree in Computer Engineering from Keio University, Japan, in 1998. He worked for NCEPU as assistant lecturer, lecturer, and associate professor from 1983 to 1994, and as instructor and associate professor for Department of Computing Science at University College of the Cariboo, Canada, from 1998 to 2001. He is now an assistant professor in School of Information Technology and Engineering (SITE) at University of Ottawa, Canada. His current research interests are on image/video processing and multimedia communications. He is a member of IEEE.



Abdulmotaleb El Saddik is an assistant professor in the School of Information Technology and Engineering (SITE) at University of Ottawa. He received his Ph.D. (Dr.-Ing.) and M.Sc. (Dipl.-Ing.) degree in Electrical Engineering and Information Technology from Darmstadt University of Technology, Germany. Dr. El Saddik is the author or coauthor of four books and more than 30 research papers in the area of software engineering development of distributed multimedia applications and shared environments. He is an Associate Editor of the ACM Journal of Educational Resources in Computing (JERIC) and Co-editor of the IEEE-Distributed Systems Online, in the areas of Collaborative Computing and Distributed Multimedia. He is a senior member of IEEE, a member of ACM, and co-founder of ACS.