An image feature based robust digital watermarking scheme

Chih-Wei Tang[1] and Hsueh-Ming Hang[2]
Department of Electronics Engineering
National Chiao-Tung University
Hsinchu, Taiwan, R.O.C.

ABSTRACT

A novel robust digital image watermarking scheme which combines image feature extraction and image normalization is proposed. The goal is to resist both geometrical and signal processing attacks. We adopt a feature extraction method called Mexican Hat wavelet scale interaction. The extracted feature points can survive various attacks such as common signal processing, JPEG compression, and geometric distortions. Thus, these feature points can be used as reference points for both watermark embedding and detection. The normalized image of a rotated image (object) is the same as the normalized version of the original image. As a result, the watermark detection task can be much simplified when it is done on the normalized image without referencing to the original image. However, because image normalization is sensitive to image local variation, we apply image normalization to non-overlapped image disks separately. The center of each disk is an extracted feature point. Several copies of one 16-bit watermark sequence are embedded in the original image to improve the robustness of watermarks. Simulation results show that our scheme can survive low quality JPEG compression, color reduction, sharpening, Gaussian filtering, median filtering, printing and scanning process, row or column removal, shearing, rotation, scaling, local warping, cropping, and linear transformation.

Keywords: Robust watermark, feature extraction, image normalization, Marr wavelet, geometric distortion.

1. INTRODUCTION

Many digital watermarking schemes have been proposed for copyright protection recently due to the rapid growth of multimedia data distribution. On the other hand, attacks have been developed to destroy watermarks. These attacks on watermarks can roughly be classified as geometric distortions and noise-like signal processing. Geometric distortions are difficult to tackle. They can induce synchronization errors between the extracted watermark and the original watermark at detection even though the watermark still exists in the watermarked image. Nowadays, several approaches that counterattack geometric distortions have been developed. These schemes can be roughly divided into invariant transform domain based, moment-based and feature extraction based algorithms.

Watermark embedded in invariant-transform based algorithm generally maintain synchronization under rotation, scaling and translation. Examples of these transforms are log-polar mapping of DFT [1][2][3][4] and fractal transform coefficients [5]. A structured template may be embedded in the DFT domain to assist watermark synchronization at detection [2][3]. To reduce visibility of template and reduce its interference to the previously embedded watermarks, the data capacity of this approach is low. A fixed structure template may be identified and destroyed easily. Watermarks embedded in DFT domain are sensitive to the other types geometrical transformation such as local warping. There is an accuracy problem associated with log-polar mapping of DFT since the interpolation process cannot be avoided during inverse transformation.

The watermark detection process is similar to the pattern recognition process in computer vision, but the original images may not be available at the watermark detector. Moments of objects have been widely used in pattern recognition. Higher order moments are more sensitive to noise and some normalization schemes have been designed to tolerate noise [6]. A watermarking system employing image normalization technique that uses moments is proposed in [7]. If the image normalization process is applied to the entire image, it would be sensitive to the cropping operation and local region distortion. Another moment based watermarking scheme [3] hides watermarks by modifying image content iteratively to

[1] u8811831@cc.nctu.edu.tw
[2] hmhang@cc.nctu.edu.tw

Security and Watermarking of Multimedia Contents IV, Edward J. Delp III, Ping Wah Wong, Editors, Proceedings of SPIE Vol. 4675 (2002) © 2002 SPIE · 0277-786X/02/$15.00

produce the mean value of several invariant moments in a predefined range. The watermark detector claims the existence of watermark by checking the mean value of these moments. This scheme can resist orthogonal transformations and general affine transformation, but it is sensitive to cropping and aspect ratio changes.

The extracted feature of image content can be used as reference points for both watermark embedding and detection [10][11][13]. In [13], Harris detector and Achard - Rouquet detector are used for feature extraction. The simulation result shows that this scheme is less effective for images with mainly textures. In [11], authors suggest retrieving feature points by the Mexican Hat wavelet scale interaction method. These points are connected to form a Voronoi diagrams for watermark embedding, and they experimentally show that it is very robust to JPEG compression [12]. Although these feature points are rotation-invariant, the embedded watermarks in the Voronoi diagrams are not rotation-invariant and thus still have to be searched in the rotated images.

In this paper, we develop a new robust watermarking scheme. This scheme combines the advantages of feature extraction and image normalization process to resist image geometric distortion and to reduce watermark synchronization problem at the same time. Section 2 describes the feature extraction method used in the proposed scheme. In section 3, the image normalization process developed for pattern recognition is briefly reviewed. Section 4 contains the description of our watermark embedding procedure. Section 5 covers the details of the watermark detection procedure. Simulation results in section 6 will show the performance of our scheme. Finally, section 7 concludes this presentation.

2. FEATURE EXTRACTION

In order to detect watermarks without using original images at detection, we look for reference points that are perceptually significant and can thus resist various types of common signal processing, JPEG compression and geometric distortions. These reference points can also act as marks for (location) synchronization between watermark embedding and detection. In this paper, we will use the term "feature points" to denote these reference points.

In our scheme, we adopt a feature extraction method called Mexican Hat wavelet scale interaction that was originally used in [12]. This feature extraction method determines the feature points by identifying the intensity changes in an image. Since significant intensity changes (edges) may occur at different scaled versions of the same image, Marr and Hildreth suggested different operators should be used at different scales for optimally detecting significant intensity changes. Mexican Hat wavelet (Marr wavelet) [14][15] is a rotation invariant wavelet. It has a circularly symmetric frequency response. The computational cost is high because this wavelet is not separable. In fact, it is the Laplacian of a Gaussian function. Similar to conventional wavelets, this wavelet analysis filter is localized at different frequencies and spatial scales (resolutions). The Mexican-Hat mother wavelet at location $\vec{x}$ is defined by (1):

$$\psi(\vec{x}) = (2 - |\vec{x}|^2)\exp(-\frac{\vec{x}^2}{2}), \tag{1}$$

where $|\vec{x}| = (x^2 + y^2)^{1/2}$ is in the two-dimensional spatial domain. And, in the spatial-frequency domain, it is defined by (2),

$$\psi(\vec{k}) = (\vec{k} \cdot \vec{k})e^{-1/2(\vec{k}\cdot\vec{k})}, \tag{2}$$

As we can see, the larger value the scale is, the finer resolution we get. The feature extraction method proposed in [17] is shown by equations (3) and (4):

$$P_{ij}(\vec{x}) = |M_i(\vec{x}) - \gamma \cdot M_j(\vec{x})|, \tag{3}$$

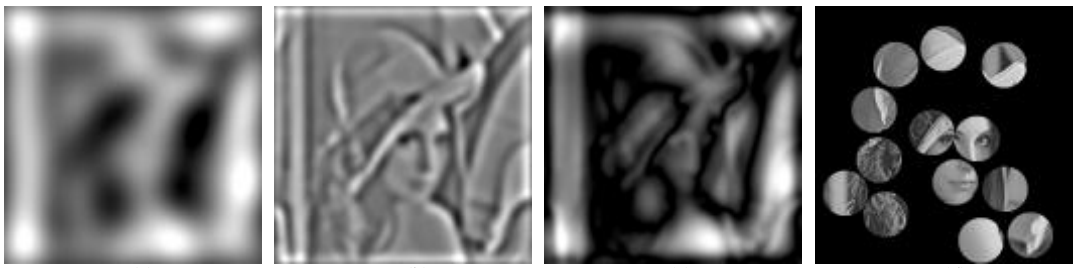
$$M_i(\vec{x}) = \langle (2^{-i}\psi(2^{-i} \cdot \vec{x})); A \rangle, \tag{4}$$

where $M_i(\vec{x})$ represents the frequency response of the Mexican hat wavelet at spatial location $\vec{x}$ of scale i, γ is a scaling parameter, $P_{ij}(\vec{x})$ is the scale interaction between two different scales i and j, A is the input image,

$\langle \cdot, \cdot \rangle$ denotes the convolution operation, and $|\cdot|$ is the absolute value function.

Our scheme is designed for both color and gray-level images. For color images, the Y component is extracted for watermark embedding. To reduce computation, the Mexican Hat wavelet filtering is calculated in the frequency domain assisted by FFT. An input image is first zero-padded to 1024x1024 in size. We avoid selecting feature points located near borders of an image. Hence, a prohibited zone along the image border is predefined. Thus, the spectral leakage effect in FFT analysis is negligible in extracting the feature points.

Examples of the filtered images of two different scales are shown in Figs. 1(a) and 1(b). The difference of these two filtered images is the Mexican Hat scale interaction image as shown in Fig. 1(c). The two scales we choose are suggested by [11][12]; that is, i=2 and j=4. Each feature point is the local maximum point inside a disk on the scale interaction image. The disk radius is chosen to be 4, which is determined experimentally. The value of γ is set to 1 in our case. The feature points located in the regions of small variance are discarded for reducing watermark visibility. The flowchart of extracting feature points using the Mexican Hat wavelet scale interaction is summarized in Fig. 2.



(a) (b) (c) (d)

Figure 1: (a) Mexican Hat wavelet filtered image with scale=2. (b) Mexican Hat wavelet filtered image with scale =4. (c) The difference image between (a) and (b). (d) The center of each disk is the selected feature point.

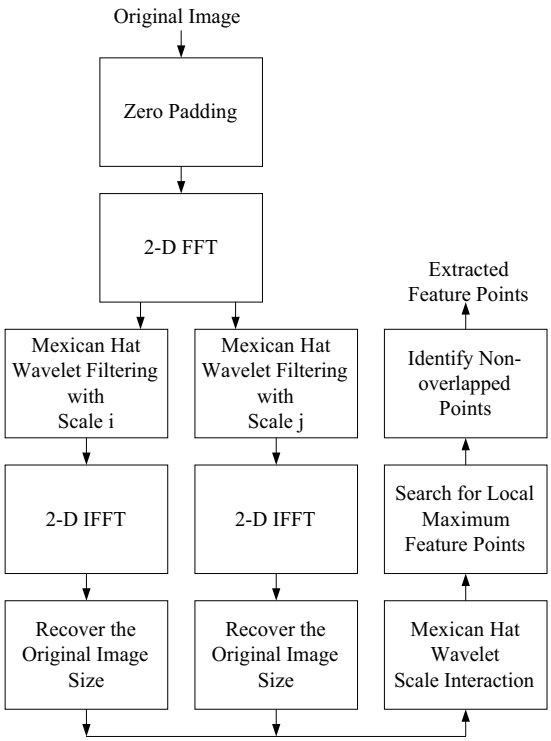


Figure 2: Feature extraction by Mexican Hat wavelet scale interaction

Among many feature extraction algorithms, we adopt the scheme proposed in [12] for several reasons. First, since the Mexican Hat wavelet scale interaction is formed by two chosen scales, it allows different degree of robustness (against distortion) by choosing proper scale parameters. Second, since local variations such as cropping or warping generally affect only a few feature points in an image, the unaffected feature points can still be used as references at detection. Third, this wavelet function is rotational-invariant. It means that most feature points may not change after image rotation. Fourth, since it is a band-limited filter, the noise sensitivity problem in feature extraction can be reduced. Finally, the extracted feature points do not shift their locations much under JPEG compression as discuss in [7].

These feature points are the centers of the disks that are to be used for watermark embedding (described in the next section). Examples of disks are shown in Fig. 1(d). Since these disks should not interfere with each other, we only select the feature points that are away from each other to create a non-overlapped disk set. In our scheme, a feature point has a higher priority for watermark embedding if it has more neighboring feature points inside its disk.

3. IMAGE NORMALIZATION

The image normalization technique developed for pattern recognition can be used for digital watermarking as suggested in [7]. Several geometric central moments are computed to transform the input image to its normalized form. The normalized image (object) of a rotated image (object) is the same as the normalized image of the original image (if no padding or cropping occurs). Since objects are rotational invariant in the normalized image, the watermark detection process can be much simplified when it is done on the normalized image. On the other hand, because image normalization is sensitive to image local variation, it performs better when applied to individual objects rather than the entire image. In our scheme, we apply image normalization process to each non-overlapped local disk separately. The centers of these disks are the extracted feature points described in section 2.

Image normalization technique is used for placing watermarks. Watermarks are not embedded to the normalized images. Only the locations of watermarks are determined on the normalized image. We do not use normalized images for watermark embedding because spatial interpolation is necessary for mapping the original image pixels to the normalized image pixels. This interpolation process induces a significant amount of distortions and thus reduces watermark detectability. The details of the image normalization process can be found in [8]. Here, we only briefly describe its computational steps as follows. The parameters below are computed once for each image disk.

1. Mean vector $[C_x C_y]^T$, where

$$C_x = \int_\Omega x f(x,y)dxdy,\ C_y = \int_\Omega y f(x,y)dxdy,\ f(x,y) = \frac{p(x,y)}{\int_\Omega p(x,y)dxdy},$$

$p(x,y)$ denotes the grey-level value at location (x,y), and Ω is the region of interest.

2. Covariance matrix $M = \begin{bmatrix} u_{20} & u_{11} \\ u_{11} & u_{02} \end{bmatrix}$, where $u_{kr} = \int_\Omega (x-C_x)^k (y-C_y)^r f(x,y)dxdy$.

3. Central moments $u_{30}, u_{21}, u_{12}, u_{03}$ of the original disk.

4. Eigenvalues λ_1, λ_2 and their associated eigenvectors $[e_{1x} e_{1y}]^T$ of M.

5. Two parameter matrices

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} \frac{c}{\sqrt{\lambda_1}} & 0 \\ 0 & \frac{c}{\sqrt{\lambda_2}} \end{bmatrix} \begin{bmatrix} e_{1x} & e_{1y} \\ -e_{1y} & e_{1x} \end{bmatrix} = \begin{bmatrix} \frac{c}{\sqrt{\lambda_1}} e_{1x} & \frac{c}{\sqrt{\lambda_1}} e_{1y} \\ -\frac{c}{\sqrt{\lambda_2}} e_{1y} & \frac{c}{\sqrt{\lambda_2}} e_{1x} \end{bmatrix}$$

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} \frac{c}{\sqrt{\lambda_1}} & 0 \\ 0 & \frac{c}{\sqrt{\lambda_2}} \end{bmatrix} \begin{bmatrix} e_{1x} & e_{1y} \\ -e_{1y} & e_{1x} \end{bmatrix} \begin{bmatrix} -C_x \\ -C_y \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} -C_x \\ -C_y \end{bmatrix}$$

where $c = (\lambda_1 \cdot \lambda_2)^{1/4}$.

6. Central moments for calculating rotational invariant transformation:

$$u'_{30} = a_{11}^2 a_{12} u_{21} + 3a_{11} a_{12}^2 u_{12} + a_{12}^3 u_{03}$$
$$u'_{21} = a_{11}^2 a_{21} u_{30} + (a_{11}^2 a_{22} + 2a_{11} a_{12} a_{21}) u_{21} + (2a_{12} a_{21} a_{22} + a_{22}^2 a_{21}) u_{12} + a_{12} a_{22}^2 u_{03}$$
$$u'_{12} = a_{11} a_{21}^2 u_{30} + (a_{21}^2 a_{12} + 2a_{11} a_{21} a_{22}) u_{21} + (2a_{12} a_{21} a_{22} + a_{22}^2 a_{21}) u_{12} + a_{12} a_{22}^2 u_{03}$$
$$u'_{30} = a_{21}^3 u_{30} + 3a_{21}^2 a_{22} u_{21} + 3a_{21} a_{22}^2 u_{12} + a_{22}^3 u_{03}$$

7. Tensors: $t^1 = u'_{12} + u'_{30}$, $t^2 = u'_{03} + u'_{21}$.

8. Angle: $\alpha = \tan^{-1}(-\frac{t^1}{t^2})$

9. Tensor $\bar{t}^2 = -t^1 \sin\alpha + t^2 \cos\alpha$

10. If $\bar{t}^2 < 0$ then $\alpha = \alpha + \pi$.

Finally, the normalized image is computed from the original image based on the following coordinate transformation:

$$\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \frac{c}{\sqrt{\lambda_1}} & 0 \\ 0 & \frac{c}{\sqrt{\lambda_2}} \end{bmatrix} \begin{bmatrix} e_{1x} & e_{1y} \\ -e_{1y} & e_{1x} \end{bmatrix} \begin{bmatrix} x - C_x \\ y - C_y \end{bmatrix},$$

where (x, y) is the original disk coordinates, and $(\bar{x}, \bar{y})$ is the normalized disk coordinates.

After the normalized disk image is generated, a rectangular window used to hold watermarks is constructed as follows. We select two 32x32 blocks in each (original) image disk for watermark embedding. The locations of these 32x32 blocks are decided on the normalized image disks. Two ordered points A and B are chosen at integer coordinates inside the normalized image disk as shown in Fig. 3 (a). The locations of these two points are generated secretly, and the watermark detector has to know the locations. The locations of A and B are chosen close to the outer circle of the normalized disk and the distance between these two points must be 32. Points a and b located on the original image are the inverse normalized mappings of A and B as shown in Fig. 3(b). Points a and b are connected to form a line segment $\overline{ab}$. Then, 31 line segments parallel to $\overline{ab}$ are created running towards the center of the disk. Each line segment contains 32 pels. Thus, $\overline{ab}$ and its 31 parallel line segments form a 32x32 block on the original image as shown in Fig. 3 (c).

Since the 32 points that a line segment passes through do not always have integer coordinates, we choose 32 integer-coordinate pels nearest to the line segment to form the discrete-grid line segment as shown in Figs. 4(a) and 4(b). In Fig. 4, the crossing points of grid represent integer-coordinate pels on the original image (disk). If the absolute value of the slope of a line segment is less than 1, its discrete-grid approximation is constructed along the horizontal direction as shown in Fig. 4(a). Otherwise, the vertical direction is used as shown in Fig. 4(b).

Two 32x32 blocks are selected for each disk as shown in Fig. 3 (d). To reduce the impact on feature point shift due to watermark embedding, these blocks should not include the disk center (feature point). All the location information of these

two blocks is decided on the normalized image disk. Then, the inverse normalization transform maps the selected blocks back to original image disk. After the coordinates of A and B are determined, their symmetric pels with respect to the disk center are labeled as C and D (Fig. 3(a)). Next, the corresponding pels c and d on the original image disk are computed by the inverse normalization transformation. All the selected blocks on image Lena are as shown in Fig. 5.

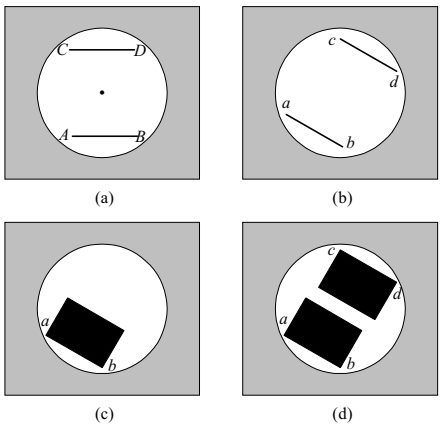


Figure 3: (a) Two ordered points A and B on the normalized image disk. (b) Two corresponding points a and b on the original image disk. (c) A 32x32 block is constructed on the original image disk. (d) Two symmetric 32x32 blocks on the original image disk are formed.

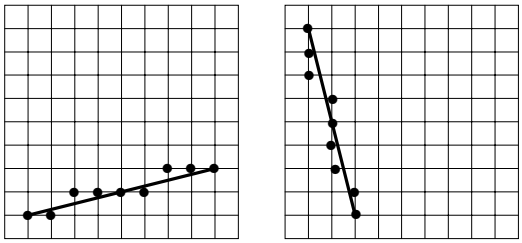


Figure 4: The crossing points of the grid represent the integer pel locations on the original disk. (a) If the slope (absolute value) of a line segment is less than or equal to 1, the integer pels closest to the line segment horizontally are chosen to form the data line segment. (b) If the slope (absolute value) of a line segment is greater than 1, the integer pels closest to the line segment vertically are chosen to form the data line segment.

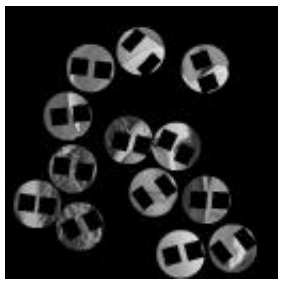


Figure 5: Each disk contains two 32x32 blocks for watermark embedding (Lena).

4. WATERMARK EMBEDDING SCHEME

Our watermark is designed for copyright protection. We view all blocks as independent communication channels. To improve the robustness of transmitted information (watermark bits), all channels carry the same copy of the chosen watermark. The transmitted information passing through each channel may be disturbed by different types of transmission

noise due to intentional and unintentional attacks. At detection, we claim the existence of watermark if at least two copies of the embedded watermark are correctly detected.

The watermark embedding process is outlined in Fig. 6. At beginning, the feature extraction method generates reference centers of disks for watermark embedding and detection. We then perform the image normalization technique on disks and select two 32x32 blocks on each disk to reduce the watermark synchronization problem at detection. Next, 2-D FFT is applied to these 32x32 blocks and the watermark is embedded in the transform domain. Last, the watermarked blocks are 2-D IFFT converted back to the spatial domain to replace the original image blocks. The DFT domain is chosen for embedding watermarks because the theoretical analysis in [16] shows that in fighting against the compression noise the watermark bits embedded in the DFT domain are more robust than those embedded in the DCT domain and in the subband transform domain.

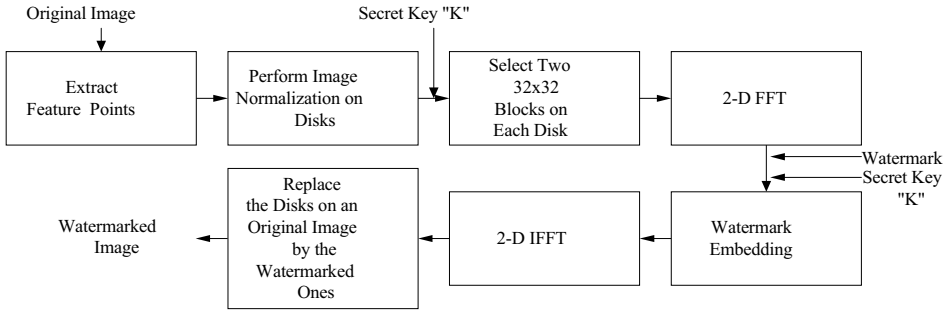


Figure 6: Watermark embedding scheme.

The procedure of selecting and modifying DFT coefficients for watermark embedding is illustrated below. First, one 32x32 selected block described in Section 3 is transformed by FFT. Then, several middle frequency coefficients in the DFT domain are selected randomly. Middle frequency components are generally more robust in resisting compression processes. A modified version of [3] is used to embed watermark bits into DFT coefficients. A selected pair, (x_i, y_i) and $(-y_i, x_i)$, 90° apart, located on the upper half DFT plane (Fig. 7) are modified to satisfy

$$F'(x_i, y_i) - F'(-y_i, x_i) \geq \alpha \text{ if } wm_i = 1$$
$$F'(x_i, y_i) - F'(-y_i, x_i) \leq -\alpha \text{ if } wm_i = 0,$$

where $F'(x_i, y_i)$ and $F'(-y_i, x_i)$ are the altered coefficients at locations (x_i, y_i) and $(-y_i, x_i)$ in the DFT transform domain, α is the watermark strength, and wm_i is the binary watermark bit, which is either 0 or 1. If the original amplitude difference between points (x_i, y_i) and $(-y_i, x_i)$ is greater than α, no change is needed. Also, to produce a real-valued image after DFT spectrum modification, the symmetric points on the lower half DFT plane have to be altered to the exact same values, too. The higher value of α and the longer watermark sequence length would increase the robustness of the watermarking scheme. Hence, there is a tradeoff between robustness and transparency. In our case, we embed 16 bits in each 32x32 block.

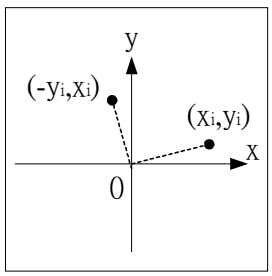


Figure 7: Two points (x_i, y_i) and $(-y_i, x_i)$, 90° apart, on the upper half DFT plane are used for embedding one watermark bit.

A secret key appears twice in Fig. 6. This secret key is also known to the watermark detector. This secret key is used as the seed for generating random numbers for two purposes. (1) They specify the locations of the 32x32 blocks used in embedding watermarks, and (2) they specify the frequencies of the DFT coefficients used to hide watermark bits.

5. WATERMARK DETECTION SCHEME

The system block diagram of watermark detection scheme is shown in Fig. 8. At watermark detection, the feature (reference) points are first extracted. The feature extraction process is similar to that used in the watermark embedding process. All the extracted feature points are candidate locations of embedded bits. The watermark detector does not need the original image. Image contents are altered slightly by the embedded marks and perhaps by attacks too. The locations of extracted feature points at detection may thus be shifted by several pixels. In addition, some of the original feature points may fail to show up at detection. If the feature point shift is small, the embedded watermark blocks can still be extracted correctly.

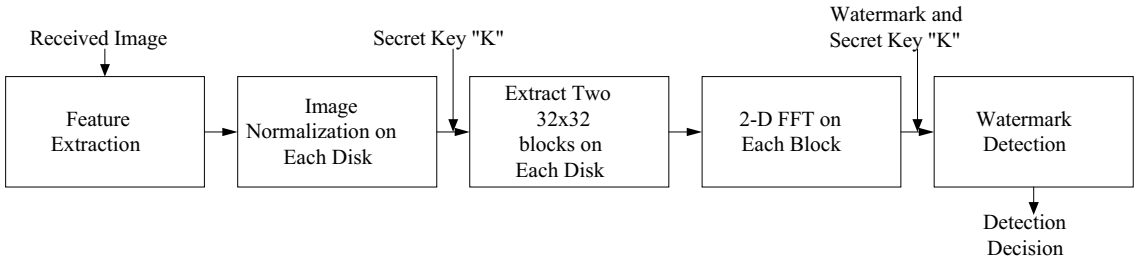


Figure 8: Watermark detection scheme.

Image normalization process is applied to all the disks centered at the extracted candidate reference points. Two 32x32 blocks are extracted from each disk using the secret key that specifies the block extraction location. In each 32x32 DFT blocks, 16 watermark bits are extracted from the DFT components specified by the same secret key used in the embedding process. For an extracted pair of DFT coefficients, (x_i, y_i) and $(-y_i, x_i)$, the embedded watermark bit is determined by the following formula,

$$wm_i = \begin{cases} 1 & \text{if } F(x_i, y_i) - F(-y_i, x_i) \geq 0 \\ 0 & \text{if } F(x_i, y_i) - F(-y_i, x_i) < 0 \end{cases}.$$

The extracted 16-bit watermark sequence is compared to the original embedded watermark. To avoid detecting a fake watermark, we adopt a false detection criterion. For a randomly selected coefficient pair in an unwatermarked image block, we assume the extracted watermark bit is equally 0 or 1. Therefore, the cumulative probability distribution function of false detection (no watermark embedded but detected having one) is

$$P_{\text{False detection}} = \sum_{r_1=T_1, r_2=T_2}^{r_1=n, r_2=n} \left(\left(\frac{1}{2}\right)^n \cdot \left(\frac{n!}{r_1!(n-r_1)!}\right)\right) \cdot \left(\left(\frac{1}{2}\right)^n \cdot \left(\frac{n!}{r_2!(n-r_2)!}\right)\right),$$

where 1/2 represents the equal probability of bits 0 and 1, n is the length of the watermark sequence, r_1 and r_2 are the numbers of correct bits in two blocks of the same disk. $P_{\text{False detection}}$ has to be upper-bounded by an acceptable threshold. In our case, n is 16. One disk is asserted to contain a valid watermark if r_1 and r_2 satisfy the following two conditions: (1) $r_1 \geq 10, r_2 \geq 10$, and (2) $r_1 + r_2 \geq 24$. Under these two conditions, the $P_{\text{False detection}}$ is less than 0.25%.

6. SIMULATION RESULTS

We test the proposed watermarking scheme on images Lena, Baboon and Peppers. They are pictures of size 512x512 as shown in Figs. 9 (a), (b), and (c). We use StirMark 3.1[17] to test the robustness of our scheme. Its generated attacks can roughly be classified into two categories: common signal processing and geometric distortions. The difference images between the original images and the watermarked images in the spatial domain are magnified by a factor of 30 and are

shown in Figs. 10 (a), (b), and (d). The PSNR value between the original and the watermarked images are 46.14 dB, 45.70 dB, and 48.29 dB for Lena, Baboon and Pepper, respectively. Because of their small amplitudes, the embedded watermarks are invisible by subjective inspection. In our simulations, the parameters are set as follows. The radius of disk on the normalized images is 45. On each disk, two 32x32 blocks are chosen for watermark embedding. In each 32x32 square, the embedded 16 frequencies (of the DFT coefficients) are located in the shaded area of Fig.11. All blocks are embedded with the same 16 bits watermark information to achieve a strong mark. The watermark strength α is set to 20 for a compromise of robustness and invisibility. As shown in Table 1, our scheme can resist JPEG up to a quality factor of 30. The JPEG compression quantization step size used in StirMark is defined by

$$Scale = \begin{cases} 5000 / quality & \text{, if quality} < 50 \\ 200 - quality \times 2 & \text{, otherwise.} \end{cases}$$

$$QuanStepSize[i] = (BasicQuanMatrix[i] \times Scale + 50) / 100. \tag{5}$$

Our scheme performs well under common signal processing such as median filtering, color quantization, 3x3 sharpening and Gaussian filtering as shown in Table 1. It can also resist the combinational attacks of common signal processing and JPEG compression at a quality factor of 90. The operations of some common signal processing used in StirMark 3.1 are outlined below. Color quantization is similar to that in GIF compression. The 3x3 Gaussian filter matrix is $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$. The 3x3 spatial sharpening filter matrix is $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$.

The performance of the proposed scheme under geometric distortions is shown in Table 2. We can see that our scheme survives row and column removal, 10% centered cropping, and up to 5% shearing in x or y direction. Rotation of small angles with cropping does not fail our scheme. But, it is still sensitive to global image aspect ratio changes due to the feature location shifts. It can also survive combinational attacks of general geometric distortions and JPEG compression as shown in Table 2. In fact, the correctness of watermark detection under geometric distortions strongly depends on the disk locations. For example, if the reference point of an image disk is located at the border of an image, this point will move or even be removed due to cropping attacks. As a result, this disk location cannot be correctly identified. Rotation can lead to a similar effect too.

In most cases, the detection performance of the Lena image is better than that of the Baboon image. The reason is that Baboon image has deeper and larger textured areas. In the case of Baboon, many fake reference points (feature points) show up and the true reference points shift quite significantly after attacks. The correct detection rate is thus lower.

In addition to the geometric distortions in StirMark 3.1, we apply local warping on the eyes and mouth of Lena as shown in Fig. 12 (a). The extracted disks at detector are shown in Fig. 12 (b). The simulation result shows that watermark can still be detected quite reliably.

The PSNR value (comparison between the watermarked image and the attacked images) in Table 3 is computed using (6).

$$PSNR = 10 \log_{10} \frac{N^2 \max_i X_i}{\sum_{i=1}^{N} (X_i - X'_i)^2}, \tag{6}$$

where N is the image size, i is the index of each pixel, and X_i and X'_i are the gray level value of the original and the processed pixels. Certain attacks produced by StirMark 3.1 have low PSNR values. Indeed, the visual artifacts can be observed on these images. Hence, they may not be useful in practice.

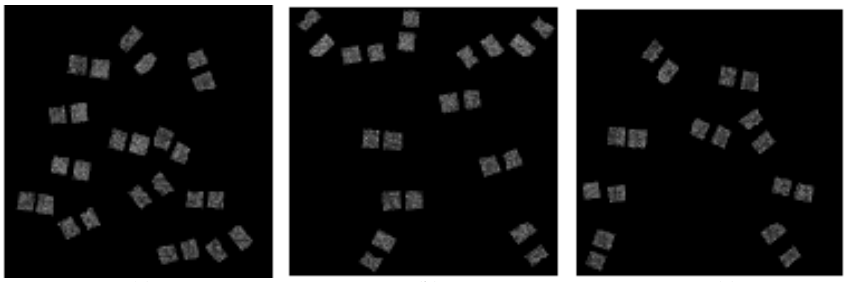
7. CONCLUSIONS

In this paper, a digital image watermarking scheme is designed to survive both geometrical and signal processing attacks. There are three key elements in our scheme: reliable image feature points, image normalization and DFT domain bits embedding. No reference images are needed at the detector. Geometric synchronization problem between the watermark embedding and detection is overcome by using visually significant points as reference points. In addition, the image normalization technique that is invariant under rotation can greatly reduce the watermark search space even if the image is geometric distorted. The simulation results show that the proposed watermarking scheme performs well under mild geometric distortion and common signal processing. Furthermore, the embedded watermark can resist the composite attacks of JPEG compression together with geometric distortions/signal processing.

The performance of our scheme can be further improved if the feature points can be even more robust. Thus, one direction of future research can be the search of more stable feature points and/or more reliable extraction algorithms under severe geometric distortions.

(a) (b) (c)

Figure 9: Original images: (a) Lena, (b) Baboon, and (c) Peppers.



(a) (b) (c)

Figure 10: The difference image between the original image and the watermarked image. The magnitude in display is amplified by a factor of 30. (a) Lena, (b) Baboon, and (c) Peppers.

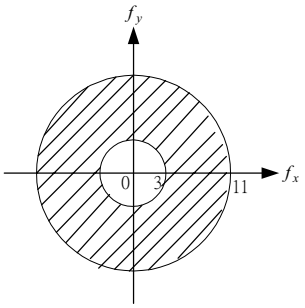


Figure 11: The watermarked coefficients are chosen from the shaded area.

(a) (b)

Figure 12: (a) Local warping is applied to watermarked image Lena. (b) Watermark detection result for (a). Ten watermarked disks are correctly detected (among the 13 original disks).

Attacks	Lena	Baboon	Pepper
Watermarked image	13/13	10/11	9/9
Median filter 2x2	7/13	6/11	4/9
Median filter 4x4	4/13	4/11	4/9
Sharpening 3x3	9/13	4/11	9/9
Color quantization	10/13	4/11	6/9
Gaussian filtering 3x3	10/13	8/11	6/9
JPEG 80	11/13	9/11	7/9
JPEG 60	11/13	7/11	7/9
JPEG 40	8/13	5/11	4/9
Median filter 2x2+JPEG90	6/13	6/11	4/9
Median filter 4x4+JPEG90	2/13	1/11	4/9
Sharpening 3x3+JPEG90	8/13	2/11	9/9
Gaussian filtering 3x3 + JPEG90	10/13	8/11	6/9

Table 1: Watermark detection on watermarked images under common signal processing attacks. (The numbers shown are correctly detected watermarked disks/Original watermarked disks)

Attacks	Lena	Baboon	Pepper
Removed 1 row and 5 columns	8/13	6/11	7/9
Removed 5 rows and 17 columns	2/13	3/11	4/9
Centered cropping 5% off	5/13	2/11	4/9
Centered cropping 10% off	4/13	2/11	3/9
Shearing-x-1%-y-1%	10/13	5/11	2/9
Shearing-x-0%-y-5%	5/13	3/11	3/9
Shearing-x-5%-y-5%	1/13	2/11	0/9
Rotation 1+Cropping+Scale	2/13	4/11	1/9
Rotation 1+Cropping	5/13	3/11	4/9
Rotation 2+Cropping	3/13	1/11	2/9
Rotation 5+Cropping	3/13	0/11	1/9
Linear geometric transform (1.007,0.01,0.01,1.012)	9/13	4/11	6/9
Linear geometric transform (1.010,0.013,0.009,1.011)	7/13	4/11	4/9
Linear geometric transform (1.013,0.008,0.011,1.008)	7/13	5/11	3/9
Removed 1 rows 5 columns + JPEG70	9/13	6/11	8/9
Removed 5 rows 17 columns + JPEG70	4/13	3/11	3/9
Centered cropping 5% + JPEG70	4/13	2/11	4/9
Centered cropping 10% + JPEG70	4/13	2/11	3/9
Shearing-x-1%-y-1%+JPEG70	10/13	4/11	2/9
Shearing-x-0%-y-5%+JPEG70	3/13	3/11	3/9
Rotation 1+Cropping+Scale+JPEG70	3/13	4/11	1/9
Rotation 1+Cropping+JPEG70	7/13	3/11	3/9
Rotation 2+Cropping+JPEG70	2/13	1/11	2/9
Rotation 5+Cropping+JPEG70	3/13	0/11	2/9
Linear geometric transform (1.007,0.01,0.01,1.012) + JPEG70	8/13	3/11	5/9
Linear geometric transform (1.010,0.013,0.009,1.011) +JPEG70	9/13	5/11	4/9

Linear geometric transform (1.013,0.008,0.011,1.008) +JPEG70	9/13	5/11	2/9

Table 2: Watermark detection on watermarked images under geometric distortions.
(The numbers shown are correctly detected watermarked disks/Original embedded watermarked disks)

Attacks	Lena	Baboon	Pepper
Median filter 2x2	28.55	22.01	31.08
Median filter 4x4	23.69	18.55	25.24
Sharpening 3x3	22.17	14.23	27.78
Color quantization	7.77	5.82	7.51
Gaussian filtering 3x3	33.69	24.48	36.65
JPEG 80	38.03	31.83	43.89
JPEG 60	35.95	28.39	40.94
JPEG 40	34.65	26.62	39.02

Table 3: PSNR value (noise = difference between the watermarked image and the attacked images)

ACKNOWLEDGMENTS

This work is partially supported by the Lee & MTI Center for Networking Research at National Chiao Tung University, Taiwan.

REFERENCES

1. S. Bhattacharjee and M. Kutter, "Compression tolerant image authentication", *IEEE International Conference on Image Processing*, vol. 1, pp. 435-439, 1998.
2. C. Y. Lin, M. Wu et al., "Rotation, scale and translation resilient public watermarking for images," In *Security and Watermarking of Multimedia Contents II, Proceedings of SPIE,* vol. 3971, 2000.
3. S. Pereira, J. J. K. ÓRuanaidh, and F. Deguillaume, "Template based recovery of Fourier-based watermarks using log-polar and log-log maps," *IEEE International Conference on Multimedia Computing and Systems*, vol. 1, pp. 870-874, 1999.
4. S. Pereira and T. Pun, "Robust template matching for affine resistant image watermarks," *IEEE Transactions on Image Processing*, vol.9, no.6, June 2000.
5. Ó Ruanaidh and T. Pun, "Rotation, scale and translation invariant digital image watermarking", *IEEE International Conference on Image Processing Proceedings*, 1997.
6. Z. Ni, E. Sung and Y. Q. Shi, "Enhancing robustness of digital watermarking against geometric attack based on fractal transform", *IEEE International Conference on Multimedia and Expo.*, vol. 2, pp. 1033-1036, 2000.
7. M. Gruber and K. Y. Hsu. "Moment-based image normalization with high noise-tolerance," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, Feburary 1997.
8. M. Alghoniemy and A. H. Tewfik "Geometric distortion correction through image normalization," *IEEE International Conference on Multimedia and Expo*, vol. 3, pp. 1291-1294, 2000.
9. S. C. Pei and C. N. Lin. "Image normalization for pattern recognition," *Image and Vision Computing,* vol. 13, no. 10 December 1995.
10. M. Alghoniemy and A. H. Tewfik, "Image watermarking by moment invariants," *IEEE International Conference* on *Image Processing Proceedings*, vol. 2, pp. 73–76, Jan. 2001.
11. A. Nikolaidis and I. Pitas, "Robust watermarking of facial images based on salient geometric pattern matching", *IEEE Transactions on Multimedia*, vol. 2, no. 3, September 2000.
12. M. Kutter, S. K. Bhattacharjee, and T. Ebrahimi. "Towards second generation watermarking schemes," *IEEE International Conference on Image Processing Proceedings*, vol. 1, pp. 320-323, 1999.
13. P. Bas, J-M. Chassery and B. Macq, "Robust watermarking based on the warping of pre-defined triangular patterns," In *Security and Watermarking of Multimedia Contents II, Proceedings of SPIE*, vol. 3971, 2000.
14. J.-P. Antoine and P. Vandergheynst, "Two-dimensional directional wavelets in image processing," *International Journal of Imaging and Systems and Technology*, vol. 7, pp. 152-165, 1996.
15. D. Marr, *Vision* (Freeman, San Francisco), pp. 54-61, 1982.
16. M. Ramkumar and A. N. Akansu, "Information theoretic bounds for data hiding in compressed images", *IEEE Second Workshop on Multimedia Signal Processing*, pp.267-272, 1998.
17. Stirmark, http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark/