Chapter 6

A Robust Feature-Based Digital Image Watermarking Scheme

6.1 Introduction

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 during the detection process, 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 domains generally maintain synchronization under rotation, scaling and translation. Examples of these transforms are given in subsection 3.4.1. The watermark detection process is similar to the pattern recognition process in computer vision, but the original images may not be available to the watermark detector. Moments of objects have been widely used in pattern recognition. The discussion of moment-based watermarking techniques is found in subsection 3.4.2. On the other hand, the extracted feature of image content can be used as reference points for both watermark embedding and detection as

illustrated in subsection 3.4.3.

In this chapter, we develop a robust watermarking scheme. This scheme combines the advantages of feature extraction and image normalization to resist image geometric distortion and to reduce watermark synchronization problem at the same time. Section 6.2 describes the feature extraction method used in the proposed scheme. In Section 6.3, the image normalization process developed for pattern recognition is briefly reviewed. Section 6.4 contains the description of our watermark embedding procedure. Section 6.5 covers the details of the watermark detection procedure. Simulation results in Section 6.6 will show the performance of our scheme. Finally, Section 6.7 concludes this presentation.

6.2 Feature Extraction

To detect watermarks without access to the original images, we look for reference points that are perceptually significant and can thus resist various types of common signal processing, such as 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. This method was originally used in [62][66][84]. It 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 that different operators should be used at different scales for optimally detecting significant intensity changes. The Mexican Hat wavelet (Marr wavelet) [33][34] 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 negative Laplacian of a Gaussian function. The wavelet analysis filter is localized at different frequencies and spatial scales (resolutions). The Mexican-Hat mother wavelet at location $\vec{x}$ is defined by (6.1):

$$\Psi(\vec{x}) = (2 - \|\vec{x}\|^2) e^{-\|\vec{x}\|^2/2}, \tag{6.1}$$

where $\|\vec{x}\| = (x^2 + y^2)^{1/2}$. The two-dimensional Fourier transform of $\hat{\psi}(\vec{x})$ is given by

$$\hat{\psi}(\vec{k}) = \|\vec{k}\|^2 e^{-\|\vec{k}\|^2/2}, \tag{6.2}$$

where $\vec{k}$ represents the 2-D spatial-frequency. The feature extraction method proposed in [62][66] uses the following quantities:

$$P_{ij}(\vec{x}) = \left| M_i(\vec{x}) - \gamma \cdot M_j(\vec{x}) \right|, \tag{6.3}$$

$$M_i(\vec{x}) = (2^i \cdot \Psi(2^i \cdot \vec{x})) * A, \tag{6.4}$$

where $M_i(\vec{x})$ represents the response of the Mexican Hat wavelet operator 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, and "$*$" denotes the convolution operation.

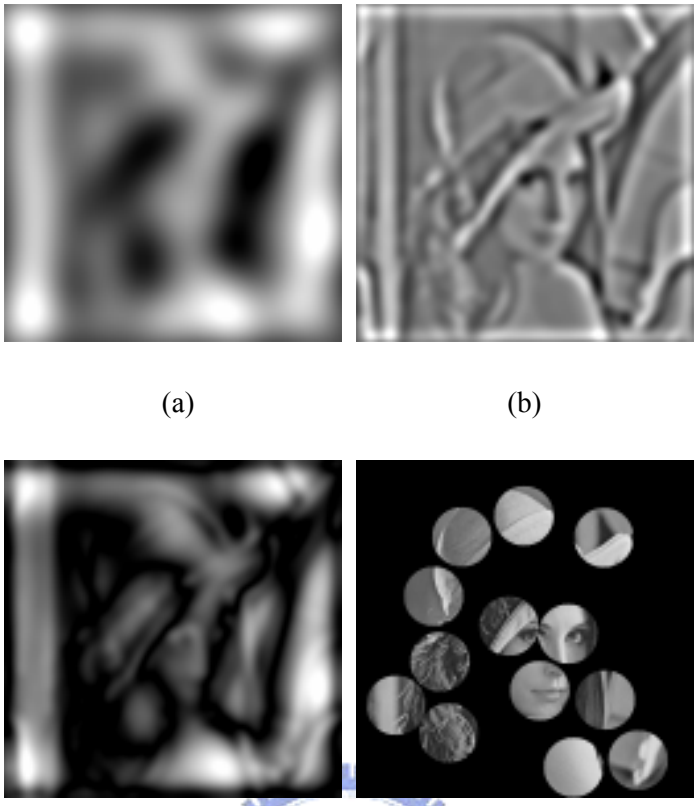
Our scheme is designed for both color and gray-level images. For color images, the Y component is extracted for watermark embedding. The Mexican Hat wavelet filtering is implemented in the frequency domain using the FFT. An input image is first zero-padded to 1024×1024 in size. We avoid selecting feature points located near borders of an image. Hence, a prohibited zone along the image border is predefined. Thus, border effects are negligible in extracting the feature points.

Examples of filtered images at two different scales are shown in Figs. 6.1(a) and 6.1(b). The difference of these two filtered images is the Mexican Hat scale

interaction image (with γ =1), shown in Fig. 6.1(c). The two scales we choose are suggested by [62][66]; that is, i =2 and j =4. Feature points are defined as local maxima inside disks in the scale interaction image. The disk radius is chosen to be 45, which is determined experimentally. Feature points located in regions of small variance are discarded for reducing watermark visibility. A flowchart of the feature extraction method is given in Fig. 6.2.

Among the many feature extraction algorithms proposed in the literature, we have adopted the scheme proposed in [62][66] for several reasons. First, since the Mexican Hat wavelet scale interaction is formed by two scales, it allows different degrees 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 during the detection process. Third, this wavelet function is rotationally-invariant. It means that most feature points may not change after image rotation. Fourth, since the Mexican Hat wavelet is essentially band-limited, the noise sensitivity problem in feature extraction can be reduced. Finally, the extracted feature points do not shift their locations much under high-quality JPEG compression as discussed in [66].

These feature points are the centers of the disks that are to be used for watermark embedding (as described in the next section). Examples of disks are shown in Fig. 6.1(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.



(a) (b)

(c) (d)

Fig. 6.1. (a) Mexican Hat wavelet filtered image at scale i=2. (b) Mexican Hat wavelet filtered image at scale i=4. (c) The difference image between (a) and (b). (d) The center of each disk is a feature point.

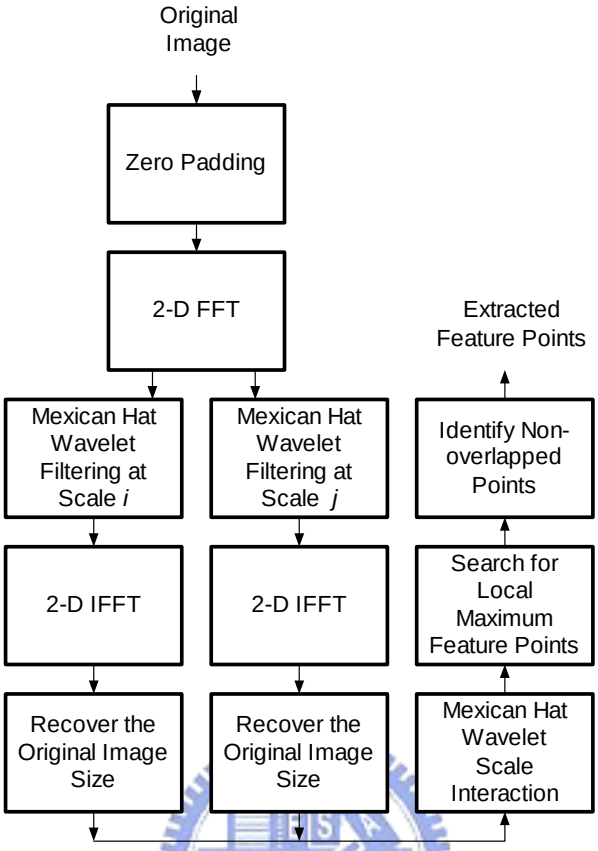


Fig. 6.2. Feature extraction by Mexican Hat wavelet scale interaction.

6.3 Image Normalization

The image normalization technique developed for pattern recognition can be used for digital watermarking as suggested in [58]. 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 rotationally invariant in the normalized image, the watermark detection process can be much simplified when it is applied to the normalized image. On the other hand, because image normalization is sensitive to local image variations, detection is more accurate when applied to individual objects rather than the entire image. In our scheme, we apply the image

normalization process to each non-overlapped local disk separately. The centers of these disks are the extracted feature points described in Section II.

Image normalization technique is used for selecting the location of the watermarks. However, watermarks are not embedded in the normalized images. This is because spatial interpolation is necessary for mapping the original image pixels to the normalized image pixels, and vice-versa. 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 [85]. Here, we only briefly describe its computational steps. The parameters below are computed once for each image disk.

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

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

where $p(x,y)$ denotes the gray-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 affine transformation coefficient 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 $c = (\lambda_1 \lambda_2)^{1/4}$.

6) Central moments for calculating rotational invariant transformation:

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

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

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

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

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

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

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

9) 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. The normalized image object is insensitive to translation, scaling and rotation of the original image object [85].

After coordinate transformation, each disk becomes a disk. Rectangular windows used to hold watermarks in the original image disks are constructed as follows. Two 32×32 blocks in each (original) image disk are chosen for watermark embedding. The locations of these 32×32 blocks are determined through the use of the normalized image disk. Two ordered points A and B are chosen at integer coordinates inside the normalized image (disk) as shown in Fig. 6.3(a). The locations of these two points are chosen secretly but are known to the watermark detector. The locations of A and B are chosen close to the boundary of the normalized disk, and the distance between these two points is 32. Points a and b located in the original image are the inverse mapping of A and B (on the normalized image) as shown in Fig. 6.3(b). Usually, points of the inverse mapping of A and B do not have integer coordinates, and thus, points a and b are quantized to integers. They are connected to form a line segment $\overline{ab}$. Although the distance between points A and B is 32, the distance between a and b is generally different due to the normalization process. Therefore, $\overline{ab}$ is shortened or extended to the line segment $\overline{ab'}$, which has length 32. Usually, point b' is not the same as point b, but these two points are close. Then, 31 line segments parallel to $\overline{ab'}$ are created running towards the center of the disk. Finally, $\overline{ab'}$ and its 31 parallel line segments of length 32 form a 32×32 block in the original image as shown in Fig. 6.3(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. 6.4(a) and 6.4(b). The crossing points of grid represent integer-coordinate pels in 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. 6.4(a). Otherwise, the vertical direction is used as shown in Fig. 6.4(b).

Two 32×32 blocks are selected for each disk as shown in Fig. 6.3(d). To reduce the impact of feature point shift due to watermark embedding, these blocks should not contain the disk center (feature point). All the location information of these two blocks is determined on the normalized image (disk). After the coordinates of A and B are determined as described above, the coordinates of C and D will be the symmetric pels with respect to the symmetric center C_e (Fig. 6.3(a)). C_e is not necessary the center of the disk. Point E is the middle point of A and B. $\overline{AB}$ is perpendicular to $\overline{EC_e}$. The distance between points A and C_e is less than 45 but greater than 32. The distance between E and C_e has to be greater than 32. Next, the corresponding pels c and d in the original image disk are computed by the inverse normalization transformation. A shortened or extended line segment of $\overline{cd}$ is $\overline{c'd}$, which contains 32 pels. The blocks selected for the image Lena are shown in Fig. 6.5. Occasionally, a tiny corner (very few pels) of a 32×32 block may be outside the original image disk. If this happens, these pels are not watermarked. Another potential problem is that although the extracted feature points (center of the disk) are located in high-contrast regions, the two 32×32 selected blocks may be partly located in smooth regions. Therefore, to keep watermark imperceptibility, such a disk is not watermarked if the variance of one 32×32 block in an original image disk is small. In our experiment, there are only 8 qualified disks (Fig. 6.5) for watermark embedding although there are 11 feature points (disk centers) are extracted on the Lena image (Fig. 6.1(d)).

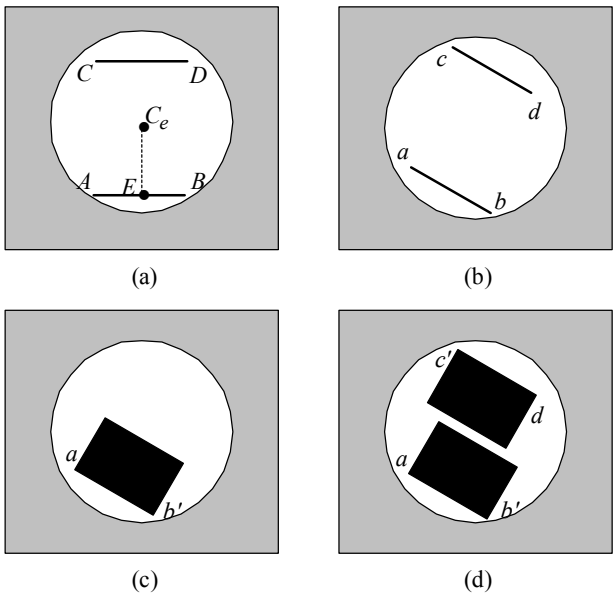


Fig. 6.3. (a) Two ordered points A and B in the normalized image (disk). (b) Two corresponding points a and b in the original image (disk). (c) A 32×32 block is constructed in the original image disk. (d) Two symmetric 32×32 blocks in the original image disk are formed.

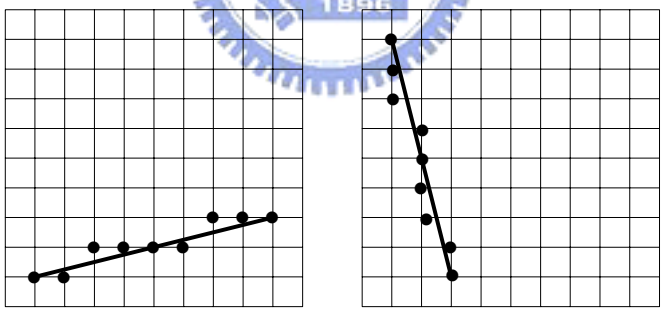


Fig. 6.4. The crossing points of the grid represent the integer pel locations on the original disk. (a) If the slope (absolute value) of a line segment is less than or equal to 1, the integer pels closest to the line segment horizontally are chosen to form the data line segment. (b) If the slope (absolute 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.

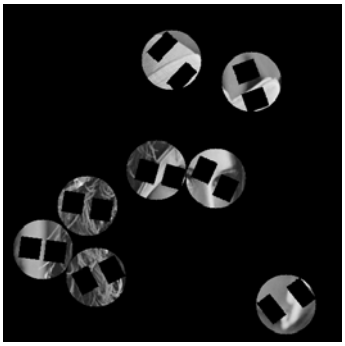


Fig. 6.5. Each disk contains two 32×32 blocks for watermark embedding (Lena).

6.4 DFT Domain Watermark Embedding

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. During the detection process, we claim the existence of watermark if at least two copies of the embedded watermark are correctly detected.

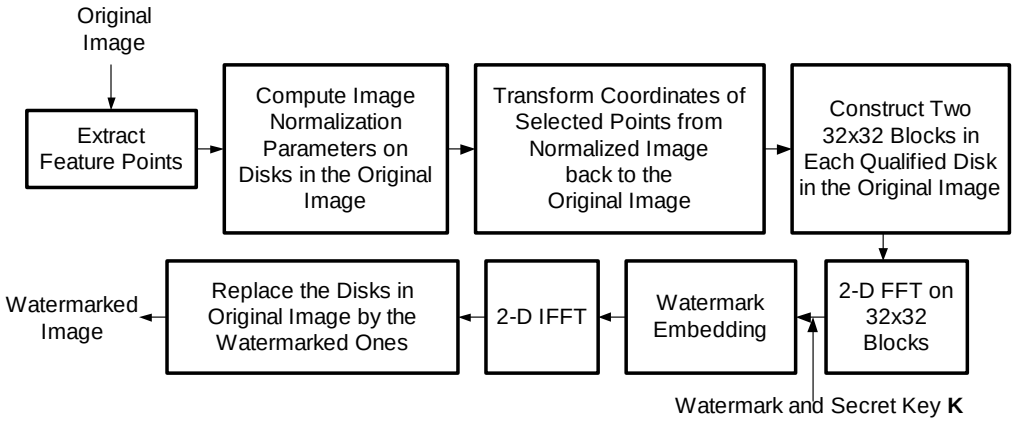


Fig. 6.6. Watermark embedding scheme.

The watermark embedding process is outlined in Fig. 6.6. First, the feature extraction method generates reference centers of disks for watermark embedding and detection. We then perform the image normalization technique on disks in the original image. The coordinate transformation coefficients between the original image disks and the normalized disk images are generated. The location of blocks in the original image for watermark embedding is determined from the normalized image. Then, coordinates of selected points are transformed from normalized image back to the original image. As a result, the watermark synchronization problem during the detection process is reduced. Next, a 2-D FFT is applied to these 32×32 blocks on each qualified disk in an original image. 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 procedure of selecting and modifying the magnitude of DFT coefficients for watermark embedding is illustrated below. First, the FFT is applied to each 32×32 selected block. Then, several middle DFT coefficients are selected according to the secret key **K**. Middle frequency components are generally more robust in resisting compression attacks. A modified version of [26] is used to embed watermark bits into DFT coefficients. Selected pairs, (x_i, y_i) and $(-y_i, x_i)$, 90° apart, located on the upper half DFT plane (Fig. 6.7) 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 magnitudes of 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. The phase of the selected DFT coefficients is not modified. If the watermark bit is 1 and 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. A larger value of α and a longer watermark sequence length would increase the robustness of the watermarking scheme. Because the 32×32 blocks are selected in the high variance image regions, typically the embedded watermark is less visible for smaller α. Hence, there is a tradeoff between robustness and transparency. In our case, we embed 16 bits in each 32×32 block.

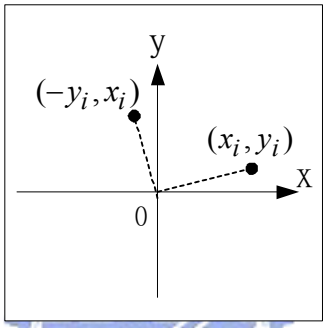


Fig. 6.7. Two points (x_i, y_i) and $(-y_i, x_i)$, 90° apart, on the upper half DFT plane are used for embedding one watermark bit.

The secret key **K** shown in Fig. 6.6 is also known to the watermark detector. This secret key is used as the seed for generating random numbers to specify the frequencies of the DFT coefficients used to hide watermark bits.

6.5 Watermark Detection

The block diagram of our watermark detection scheme is shown in Fig. 6.8. The watermark detector does not need the original image. 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. Since image contents are altered slightly by the embedded marks

and perhaps by attacks too, the locations of extracted feature points may be shifted. In addition, some of the original feature points may fail to show up during the detection process. If the feature point shift is small, the embedded watermark blocks can still be extracted correctly.

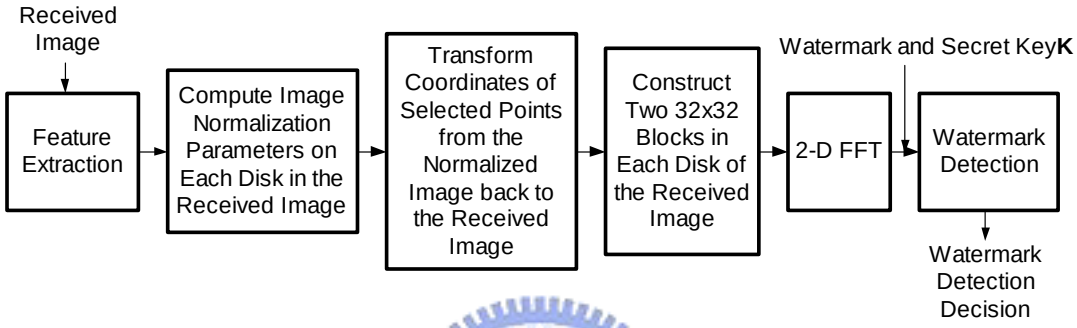


Fig. 6.8. Watermark detection scheme.

Image normalization is applied to all the disks centered at the extracted candidate reference points. Two 32×32 blocks are extracted in each disk. The locations of these 32×32 blocks are the same as those specified at watermark embedding. The coordinate transformation coefficients between the original image disk and the normalized disk image are generated. Thus, the location of blocks in the received image is determined from the normalized image, and the coordinates of the selected points are transformed from normalized image back to the received image.

In each 32×32 DFT block, 16 watermark bits are extracted from the DFT components specified by the secret key. 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,

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

where $F''(x_i, y_i)$ and $F''(-y_i, x_i)$ are the magnitudes of the selected coefficients at locations (x_i, y_i) and $(-y_i, x_i)$. The extracted 16-bit watermark sequence is then compared to the original embedded watermark for deciding a success detect.

Two kinds of errors are possible in the detector: the *false-alarm* probability (no watermark embedded but detected having one) and the *miss* probability (watermark embedded but detected having none). There is a trade-off between these two error probabilities in selecting detector parameters. Typically, reducing one will increase the other. It is rather difficult to have exact probabilistic models of these two kinds of errors. Simplified models are thus assumed in choosing the detector parameters as shown below.

We first examine the false-alarm probability. For an unwatermarked image, the extracted bits are assumed to be independent random variables (Bernoulli trials) with the same "success" probability, $P_{success}$. It is called a "success" or "match" if the extracted bit matches the embedded watermark bit. We further assume that the success probability, $P_{success}$, is ½. Let r_1 and r_2 be the numbers of matching bits in the two blocks on the same disk, and *n* the length of the watermark sequence. Then, based on the Bernoulli trials assumption, r_1 and r_2 are independent random variables with binomial distribution,

$$P_{r_1} = \left(\frac{1}{2}\right)^n \cdot \left(\frac{n!}{r_1!(n-r_1)!}\right)$$

and

$$P_{r_2} = \left(\frac{1}{2}\right)^n \cdot \left(\frac{n!}{r_2!(n-r_2)!}\right).$$

The mean values of r_1 and r_2 are both *n*/2.

A block is claimed watermarked if the number of its matching bits is greater than

a threshold. The thresholds for the two blocks on the same disk are denoted by T_1 and T_2. Clearly, T_1 and T_2 should be greater than $n/2$, the mean values of r_1 and r_2. The false-alarm error probability of a disk is, therefore, the cumulative probability of the cases that $r_1 \geq T_1$ and $r_2 \geq T_2$. In order to control the level of false-alarm probability by one adjustable parameter, a third threshold T is introduced. More precisely, the variable pairs, r_1 and r_2, shall satisfy the following two criteria simultaneously: (1) $r_1 \geq T_1$, $r_2 \geq T_2$ and (2) $r_1 + r_2 \geq T$. That is,

$$P_{False-alarm\ on\ one\ disk} = \sum_{\substack{r_1=T_1,r_2=T_2 \\ r_1+r_2\geq T}}^{r_1=n,r_2=n} \left(\frac{1}{2}\right)^n \cdot \left(\frac{n!}{r_1!(n-r_1)!}\right) \cdot \left(\frac{1}{2}\right)^n \cdot \left(\frac{n!}{r_2!(n-r_2)!}\right). \quad (6.5)$$

Furthermore, an image is claimed watermarked if at least m disks are detected as "success". Under this criterion, the false-alarm probability of one image is

$$P_{False-alarm\ on\ one\ image} = \sum_{i=m}^{N} (P_{False-alarm\ on\ one\ disk})^i \cdot (1 - P_{False-alarm\ on\ one\ disk})^{N-i} \cdot \binom{N}{i}, \quad (6.6)$$

where N is the total number of disks in an image.

We can plot $P_{False-alarm\ on\ one\ mage}$ against various T values as shown in Fig. 6.9 using (6.6). The other parameters are chosen based on our experiences: n = 16, N = 10, m = 3, $T_1 = 10$ and $T_2 = 10$. The curve in Fig. 6.9 drops sharply for T > 23. It is often desirable to have a very small $P_{False-alarm\ on\ one\ mage}$. However, the selection is application dependent. We assume that $P_{False-alarm\ on\ one\ mage}$ should be less than 10^{-5}. In this case, T should be greater than or equal to 24 and at T = 24, $P_{False-alarm\ on\ one\ mage}$ is 5×10^{-6}.

We next examine the miss probability. In an attacked watermarked image, we again assume that the matching bits are independent Bernoulli random variables with equal success probability, $P_{success}$. This may not be a very accurate model but it seems to be sufficient for the purpose of selecting the detector parameters. The

success detection probability of r_1 bits in a block of n watermarked bits is

$$P_{r_1} = (P_{success})^{r_1} \cdot (1 - P_{success})^{n-r_1} \cdot \left(\frac{n!}{r_1!(n-r_1)!} \right).$$

Similarly, for the second block

$$P_{r_2} = (P_{success})^{r_2} \cdot (1 - P_{success})^{n-r_2} \cdot \left(\frac{n!}{r_2!(n-r_2)!} \right).$$

The success detection probability of a disk is the cumulative probability of all the cases that $r_1 \geq T_1$, $r_2 \geq T_2$ and $r_1 + r_2 \geq T$. That is,

$$P_{Success\ on\ one\ disk} = \sum_{\substack{r_1=T_1, r_2=T_2 \\ r_1+r_2 \geq T}}^{r_1=n, r_2=n} P_{r_1} \cdot P_{r_2} \quad . \tag{6.7}$$

Recall that an image is claimed watermarked if at least m disks watermark detected. Under this criterion, the miss probability of an image is

$$P_{Miss\ on\ on\ e\ image} = 1 - \sum_{i=m}^{N} (P_{Sucesss\ on\ one\ disk})^i \cdot (1 - P_{Sucesss\ on\ one\ disk})^{N-i} \cdot \binom{N}{i}. \tag{6.8}$$

It is difficult to evaluate the success detection probability of a watermarked bit, $P_{success}$. It depends on the attacks. For example, the distortion induced by JPEG compression cannot be modeled by a simple additive white Gaussian source. However, a "typical" success detection probability may be estimated from the experiments on real images with attacks. Because we like to see the detector performance under geometric distortion, a moderately difficult case is chosen from Table 6.2 -- image Lena under combined distortions of 1 degree rotation, cropping and JPEG compression at a quality factor of 70. The simulation is done using 10 watermarked images Lena imposed with (randomly generated) different watermarks. The selected value of P_{success} is the total number of matching bits divided by the total number of embedded bits. In this experiment, we obtain P_{success} = 0.6883. Based on

this P_{success} value, we plot the miss probability of an image for various T as shown in Fig. 6.10. In this experiment, we set again $n=16$, $N=10$, $m=3$, $T_1=10$ and $T_2=10$. The curve goes up sharply for $T > 23$. For $T_2=24$, $P_{Miss\ on\ one\ image}$ is less than 0.42. Clearly, from Figs. 6.9 and 6.10 we can see the trade-off in selecting T. Suppose that a lower false-alarm probability is our higher priority in the simulations in Section VI, T is therefore chosen to be 24 so that $P_{False-alarm\ on\ one\ mage}$ is less than 10^{-5}.

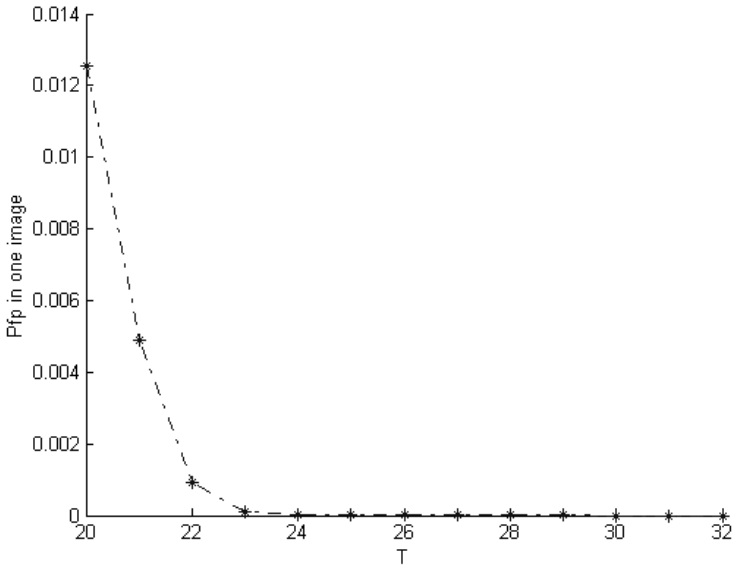


Fig. 6.9. The false-alarm probability of an unwatermarked image. The probability is generated for r_1 and r_2 satisfying the following two conditions: (1) $r_1 \geq 10$, $r_2 \geq 10$, and (2) $r_1 + r_2 \geq T$. (n =16, m=3, and N=10.)

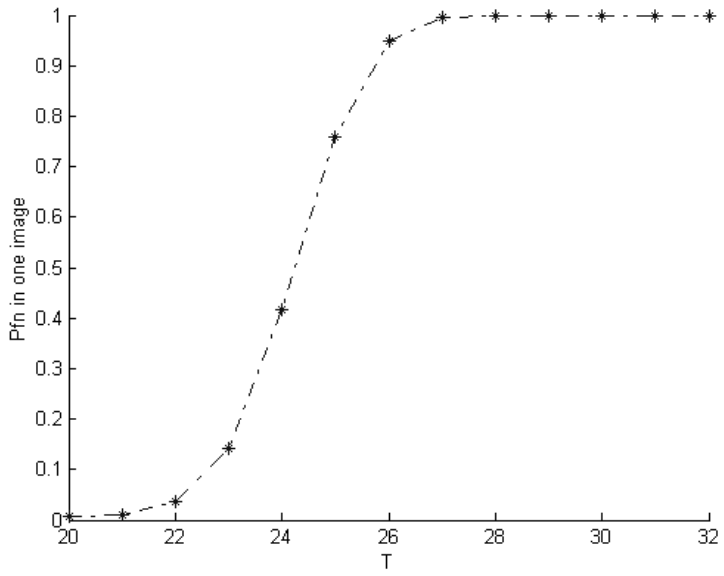


Fig. 6.10. The miss probability on image Lena. The plot is generated for r_1 and r_2 satisfying the following two conditions: (1) $r_1 \geq 10$, $r_2 \geq 10$, and (2) $r_1 + r_2 \geq T$. (n = 16, m=3, and N=10.)

6.6 Simulation Results

We test the proposed watermarking scheme on the popular test images 512×512 Lena, Baboon and Peppers. We use StirMark 3.1[78] to test the robustness of our scheme. The StirMark 3.1 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 shown in Figs. 6.11 (a), (b), and (d). The PSNR values between the original and the watermarked images are 49.42 dB, 45.70 dB, and 56.60 dB for Lena, Baboon and Peppers, respectively. Because of their small amplitudes, the embedded watermarks are invisible by subjective inspection. Recall that the radius of each disk in the normalized images is 45, and that two 32×32 blocks are chosen in each disk for watermark embedding. In each 32×32 square, the embedded 16 frequencies (of the DFT coefficients) are located within the shaded area of Fig. 6.12.

All blocks are embedded with the same 16 bits watermark. The watermark strength α is set to 20, 15 and 10 in Baboon, Lena and Peppers, respectively, for a compromise between robustness and invisibility. Since Baboon image has more texture, a strong watermark is less visible than in Lena and Peppers. The number of watermarked image disks is 11, 8 and 4 in Baboon, Lena and Peppers, respectively. The more textured the image is, the more extracted feature points the image has.

Simulation results for geometric distortions and common signal processing attacks are shown in Tables 6.1 and 6.2, respectively. The tables show the number of correctly detected watermarked disks and the number of original embedded watermarked disks. As shown in Table 6.1, our scheme can resist JPEG compression up to a quality factor of 30. The JPEG compression quantization step size used in StirMark is defined by

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

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

Our scheme performs well under other common signal processing attacks such as median filtering, color quantization, 3×3 sharpening, and Gaussian filtering. It can also resist combined signal processing and JPEG compression attacks at a quality factor of 90.

Some of the signal processing operations used in StirMark 3.1 are detailed below. Color quantization is similar to that in GIF compression. The 3×3 Gaussian filter matrix is $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$. The 3×3 spatial sharpening filter matrix is $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$. The watermark robustness against common signal processing is much improved with stronger watermark strength, but there is the tradeoff between watermark robustness and invisibility.

An additive noise attack was also applied to the watermarked image. The attacked

image is:

$$L'(x,y) = L(x,y) \cdot (1 + \beta \cdot n(x,y)) \quad ,$$

where $L(x,y)$ is the luminance pixel value of an input image at (x,y), β is a parameter that controls the strength of the additive noise, $n(x,y)$ is noise with uniform distribution, zero mean, and unit variance, and $L'(x,y)$ is the luminance pixel value of the attacked image at (x,y). In our experiment, the additive noise is visible especially in the images Lena and Peppers, when β is greater than 0.1. The watermark can be detected when β is less than 0.2. As stated in Section 6.2, the noise sensitivity problem in feature extraction is reduced due to the essentially band-limited property of Mexican Hat scale interaction scheme with proper parameter settings.

The PSNR value (comparison between the watermarked image and the attacked images) in Table 6.3 is computed by

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

where N is the image size, i is the index of each pixel, and X_i and X'_i are the gray levels of the original and the processed pixels.

The performance of the proposed scheme under geometric distortions is shown in Table 6.2. Our scheme survives row and column removal, 10% centered cropping, and up to 5% shearing in x or y direction. Combination of small rotations with cropping does not cause our scheme to fail. But, it is still sensitive to global image aspect ratio changes due to the feature location shifts. It can also survive combined geometric and high quality JPEG compression attacks, as shown in Table 6.2. 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 might be removed due to cropping attacks. As a result, this

disk location cannot be correctly identified. Rotation with cropping can have to a similar effect.

The Baboon image has deeper and larger textured areas than Lena and Peppers. In the case of Baboon, many fake reference points (feature points) may show up, and the true reference points may shift quite significantly after attacks. On the other hand, Peppers has less texture. Its true feature points may disappear following attacks.

In addition to the geometric distortions in StirMark 3.1, we have applied local warping on the eyes and mouth of Lena, as shown in Fig. 6.13(a). The extracted disks at detector are shown in Fig. 6.13(b). Since local variations generally affect only a few feature points extracted by the Mexican Hat wavelet scale interaction scheme, the feature points can still be correctly extracted for watermark detection. The watermark can still be detected quite reliably.

(a) (b) (c)

(a) (b) (c)

Fig. 6.11. The difference image between the original image and the watermarked image. The magnitudes in display are amplified by a factor of 30. (a) Lena, (b) Baboon, and (c) Peppers.

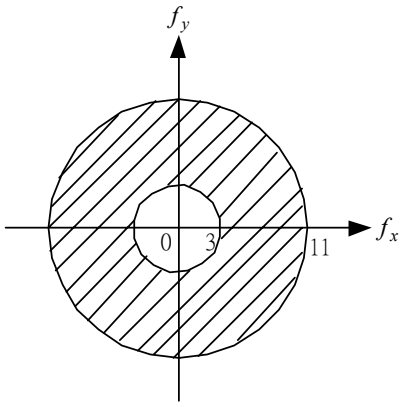


Fig. 6.12. The watermarked coefficients are chosen from the shaded area.

(a) (b)

Fig. 6.13. (a) Local warping is applied to watermarked image Lena, in the eyes and mouth area (b) Watermark detection result for (a). Seven watermarked disks are correctly detected among the original eight.

Table 6.1. Fraction of correctly detected watermark disks under common signal processing attacks.

Attacks	Lena	Baboon	Pepper
Watermarked image	7/8	10/11	4/4
Median filter 2×2	1/8	6/11	1/4
Median filter 3×3	1/8	2/11	1/4
Sharpening 3×3	4/8	4/11	4/4
Color quantization	7/8	4/11	1/4
Gaussian filtering 3×3	5/8	8/11	1/4
Additive uniform noise (scale=0.1)	5/8	6/11	4/4
Additive uniform noise (scale=0.15)	4/8	4/11	2/4
Additive uniform noise (scale=0.2)	1/8	5/11	1/4
JPEG 80	6/8	9/11	3/4
JPEG 70	7/8	11/11	3/4
JPEG 60	6/8	7/11	1/4
JPEG 50	5/8	7/11	3/4
JPEG 40	3/8	5/11	1/4
JPEG 30	2/8	4/11	0/4
Median filter 2×2 + JPEG90	2/8	6/11	0/4
Median filter 3×3 + JPEG90	1/8	1/11	1/4
Sharpening 3×3 + JPEG90	4/8	2/11	4/4
Gaussian filtering 3×3 + JPEG90	5/8	8/11	2/4

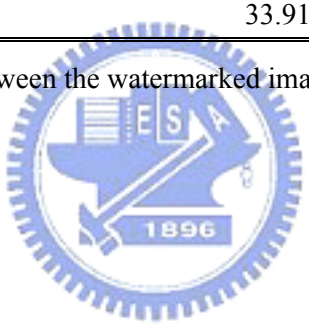
Table 6.2. Fraction of correctly detected watermark disks under geometric distortion attacks.

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

Table 6.3. PSNR values.

Attacks	Lena	Baboon	Pepper
Median filter 2×2	28.58	22.01	31.14
Median filter 3×3	31.53	24.89	31.84
Sharpening 3×3	22.24	14.23	28.08
Color quantization	7.78	5.82	7.51
Gaussian filtering 3×3	33.73	24.48	36.75
Additive uniform noise (scale=0.1)	32.04	31.40	31.77
Additive uniform noise (scale=0.15)	28.57	27.90	28.25
Additive uniform noise (scale=0.2)	26.13	25.47	25.75
JPEG 80	38.13	31.83	44.46
JPEG 70	36.92	29.71	42.67
JPEG 60	36.06	28.39	41.36
JPEG 50	35.42	27.47	40.39
JPEG 40	34.75	26.62	39.36
JPEG 30	33.91	25.69	38.06

(noise = difference between the watermarked image and the attacked images)



6.7 Summary

In this chapter, a digital image watermarking scheme was designed to survive both geometric distortion 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 invariance properties of the image normalization technique can greatly reduce the watermark search space. The simulation results show that the proposed watermarking scheme performs well under mild geometric distortion and common signal processing attacks. Furthermore, the embedded watermark can resist composite attacks of high quality JPEG compression together with geometric distortions/signal processing.

The performance of our scheme could be further improved if the feature points were 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.