Multimed Tools Appl (2007) 34:337–353
DOI 10.1007/s11042-007-0112-0

Geometrically invariant watermarking: synchronization through circular Hough transform

Hae-Yeoun Lee · Choong-hoon Lee · Heung-Kyu Lee

Published online: 2 March 2007

Abstract This paper addresses a geometrically invariant watermarking method for digital images. Most previous watermarking algorithms perform weakly against geometric distortions, which desynchronize the location for the inserted watermark. Watermark synchronization, which is a process for finding the location for watermark insertion and detection, is crucial for robust watermarking. In this paper, we propose a watermarking method that is robust to geometric distortions. In order to synchronize the location for watermark insertion and detection, we use circular Hough transform, which extracts circular features that are invariant to geometric distortions. The circular features are then watermarked using additive way on the spatial domain. Our method belongs to the category of blind watermarking techniques, because we do not need the original image during detection. Experimental results support the contention that our method is useful and considerably robust against both geometric distortion attacks and signal processing attacks as listed in Stirmark 3.1.

Keywords Geometrically invariant watermarking · Watermark synchronization · Circular Hough transform

1 Introduction

Due to the growth of network and computer technology, digital multimedia is widely used and accessed everywhere. However, digital multimedia can be copied, manipulated, and re-

H.-Y. Lee (✉)
Weill Medical College, Cornell University, 575 Lexington Av, 3F, New York, NY 10022, USA
e-mail: hytoiy@casaturn.kaist.ac.kr

C.-h. Lee
Digital Media R&D Center, SAMSUNG Electronics Co., LTD, 416 Maetan-dong, Yeongtong-gu, Suwon-si, Gyeonggi-do, South Korea

H.-K. Lee
Department of EECS, Korea Advanced Institute of Science and Technology, 373-1 Guseong-dong, Yuseong-gu, Daejeon, South Korea

produced illegally, without quality degradation and protection. Therefore, copyright protection has become a social issue.

Digital watermarking is a solution for copyright protection in which the copyright information is inserted into the contents itself. This information is used as evidence of ownership. Since Cox et al. [3] proposed spread-spectrum watermarking, there has been research conducted that is inspired by methods of image coding and compression. All of these algorithms perform robustly against signal processing attacks. Nevertheless, in blind watermarking in particular, these algorithms exhibit severe weakness to geometric distortion attacks, which desynchronize the location of the inserted copyright information and hence prevent watermark detection.

In order to resist geometric distortion attacks, watermark synchronization, which is a process of calculating the location for watermark insertion and detection, is required. There have been several studies related to watermark synchronization: the use of periodic sequences [5], the use of templates insertion [10], and the use of invariant transforms [1, 7, 11, 12]. One way to synchronize the location for watermark insertion and detection is to use the media content itself. Generally, the features of media contents represent an invariant reference for geometric distortions so that referring features can solve the problem of watermark synchronization. Section 2 will review watermarking methods that use media contents for watermark synchronization. Through this paper, we call the location for watermark insertion and detection *the patch*.

In watermark synchronization by reference to media contents, the extraction of features is important for achieving robustness of the watermark. In this paper, we propose a new geometrically invariant watermarking method that uses circular Hough transform for watermark synchronization. Through circular Hough transform, we extract circular features (circular patches) that are invariant to geometric distortions. The watermark is inserted into the circular patches in an additive way on the spatial domain. We describe the error probability of the proposed method by using the Gamma distribution model. We performed experiments to show the robustness of the proposed method with ten test images. The results support the contention that our method is useful and considerably robust against geometric distortion attacks, as well as signal processing attacks.

The following section reviews watermark synchronization methods using media contents. Section 3 describes modified circular Hough transform. In Section 4, we propose our watermarking method using circular Hough transform with the analysis of the detector performance. Section 5 shows the experimental results and Section 6 concludes.

2 Watermark synchronization using media contents

From watermark synchronization using media contents, contents are exploited to extract features. The features represent an invariant reference to distortions such that the use of media contents can solve the problem of watermark synchronization. The location of the watermark is not related to image coordinates, but image semantics [6]. This section reviews watermark synchronization methods using image features to extract the patches. We divide into two categories: feature point based synchronization and region-based synchronization.

2.1 Feature point based synchronization

Kutter et al. [6] describe a feature-based synchronization method. First, they extract feature points using a scale interaction technique based on 2D continuous wavelet. Then, they use

these points to segment the image, using a Voronoi diagram partitioning of the image such that all pixels in the image were closer to the location of the feature points. The spread-spectrum watermark is inserted into each segment separately. This method is robust to most attacks. However, it fails to synchronize the location of the watermark because the feature points from the scale interaction technique are sensitive to changes of image scale.

Bas et al. [2] suggest a synchronization method similar to that of Kutter et al. They extract feature points by applying the Harris corner detector, which uses differential features of the image. The feature points are decomposed into a set of disjoint triangles by Delaunay tessellation that is the straight line dual of the corresponding Voronoi diagram obtained by joining all pairs of points. These triangles are watermarked by the additive spread-spectrum method on the spatial domain. The weakness of this method is that the set of extracted feature points from the original image and attacked images do not match, because the Harris corner detector uses differential features that are sensitive to image noise. Therefore, the set of triangles from the feature points of the original and attacked images differ significantly and the resulting patches do not correspond.

Tang and Hang [13] use intensity-based feature extraction and image normalization for watermark synchronization. They first extract feature points using the Mexican Hat wavelet scale interaction method, which determines feature points by identifying changes in intensity in the image. Disks of fixed radius R, whose centers is the feature points, are then normalized to achieve geometric-distortion invariance. These normalized disks are watermarked on the frequency domain. Although this method works well in response to most attacks, it shows severe weakness in response to scaling distortion, because radius of the disks are fixed in such a manner that different contents are used for normalization.

2.2 Region-based synchronization

Nikolaidis and Pitas [9] describe an image-segmentation based synchronization method. By applying an adaptive k-mean clustering technique, they segment images and select several of the largest regions. The bounding rectangles of these regions are used as the patches for watermarking. The segment regions are expected to resist geometric attacks. However, image segmentation depends on image contents, so image attacks severely affect the segmentation results. Moreover, it is difficult to extract the patches when the images have complex texture.

For watermark synchronization using media contents, the extraction of the patches is important for the design of robust watermarking. When the patches differ during watermark insertion and detection, it is impossible or difficult to prove the ownership of the contents. Hence, it is important to select carefully the features to be extracted for generating the patches. We suspect that the consideration of local image characteristics during feature extraction would be helpful for generating patches that would prove effective against image distortions. Therefore, we adopt circular Hough transform to extract the patches, which considers local image characteristics and is robust to image distortions.

3 Modified circular Hough transform

In image analysis applications, Hough transform is used to recognize patterns in images, such as lines, curves, etc., that can be specified easily by a parametric equation. It is well-known that this transform is tolerant of gaps in feature boundary points and relatively unaffected by noise and image modification. Therefore, it can help in designing the patches for robust watermarking.

The basic idea of Hough transform is as follows: (1) transform feature points in the image space into those in the parametric space; then (2) extract the local maximums or peak points in the parametric space, which indicate searching objects. Each point in the image space represents a geometric model (a parametric equation) in the parametric space. Each point in the parametric space indicates searching objects, such as a line or a circle, in the image space. The dimension of the parametric space is equal to the degree of freedom of the object that we are searching.

In natural images, many objects can be approximated to circles. We apply circular Hough transform to extract circular features (their center coordinates and radii) from images and adopt these features as circular patches for watermark insertion and detection. The searching object model is a circle in this transform and defined by the following parametric (1).

$$(x-a)^2 + (y-b)^2 = r^2 \tag{1}$$

where a and b are the center coordinates of a circle and r is its radius. The parametric space is three-dimensional (a, b, and r axes) and three variables should be determined to detect a circle of arbitrary size.

Circular Hough transform is performed as follows (see Fig. 1).

1. Find feature points, such as edges and corners, in the image space (e.g., points A, B, and C).
2. For each feature point
   A. Transform each point in the image space into a geometric model in the parametric space.
   B. Accumulate boundary points of the geometric model in the parametric space.
3. Detect the local maximums, i.e. strong peak points, in the parametric space; then, they represent circular objects in the image space.

The complexity of Hough transform is related to the dimension of the parametric space, i.e. the nature of the searching object. The required time and storage space is $O(N^d)$. N is the space size and d is the dimension of the parametric space. As we use 512 by 512 images, the complexity of circular Hough transform is $O(512^3)$ and it is computationally expensive [8]. To reduce the computational burden, we use the gradient vector of feature points, edges. The direction map of feature points is first calculated from the edge maps of

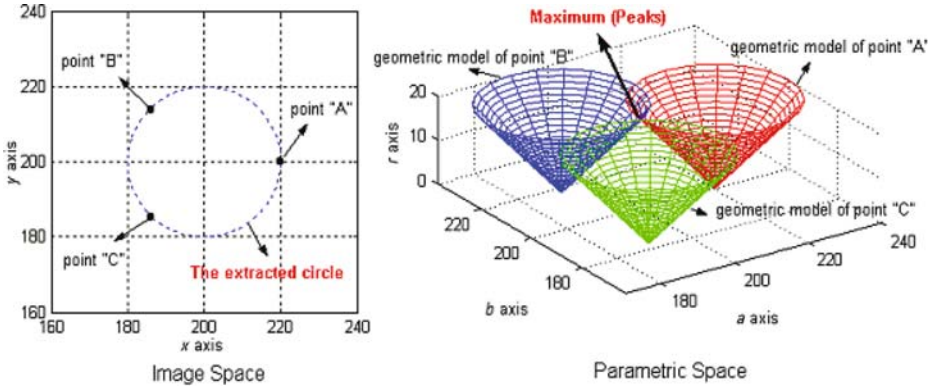


Fig. 1 Image and parametric space of *circular* Hough transform

the vertical and horizontal Sobel filter. During the accumulation of boundary points of each geometric model, we calculate a line that is perpendicular to the direction of the gradient vector in the parametric space and only consider boundary points of the geometric model near to this perpendicular line. In this way, we can reduce the number of boundary points that need to be considered.

In order to use them for watermarking purposes, we need to acquire circular features distributed over the whole image. Prior to finding circles, we apply a Gaussian filter to blur images, which increases the curvature of the edges of objects, thus making them more like circles. Furthermore, by this means, we reduce the influence of noise and increase the robustness and stability of circular features with image attacks. The radius of extracted circular features is too small to use directly, so we scale the radius of circular features by a constant factor and remove circles whose radius are too small or too large. In our experiments, we set the scale factor to 8.0. The spatially regular distribution of circular features is important for efficient watermark insertion and detection. To control the distribution of extracted features, we apply a circular neighborhood constraint used in Bas et al. [2]. Circle diameters depend on image dimension, as in (2):

$$D = \frac{w + h}{r} \tag{2}$$

where w and h denote the width and height of the image and the neighborhood size D is denoted by r.

4 Proposed watermarking method

This section describes our watermarking method. We extract circular patches by using circular Hough transform, as described in Section 3. The 2D watermark is generated and transformed into circular form for each patch. The circular watermarks are added into the patches. We first describe the watermark generation procedure and then explain watermark insertion and detection.

4.1 Watermark generation

We generate a 2D rectangular watermark that follows a Gaussian distribution, using a random number generator. To be inserted into circular patches, this watermark should be transformed into a circular shape. We consider a rectangular watermark to be polar-mapped, and polar-map inversely to assign the insertion location of the circular patches. In this way, a rotation attack is mapped as a translation into the rectangular watermark and the watermark can be detected using the correlation detector. Note that the size of the circular patches differs, so we should generate a separate circular-shaped watermark for each patch.

Let the x and y magnitudes of the rectangular watermark be denoted by M and N, respectively. The radius r of a circle is equal to the size of the circular patch. As shown in Fig. 2, we divide a circle into homocentric regions. The x- and y-axis of the rectangular watermark are polar-mapped inversely into the radius and angle directions of a circle. rM is equal to the radius of the circle r and $r\,0$ is determined relatively from rM. We set $r\,0$ as $rM/4$. For effective transformation, $r\,0$ should be larger than M/π and the difference between rM and $r\,0$ should be larger than N. If these constraints are not satisfied, the rectangular watermark must be sampled. As a result, it is difficult to transform efficiently. To increase the robustness

of the inserted watermark, we transform the rectangular watermark to be mapped to only the upper half of a circle, i.e. the y-axis of the rectangular watermark is scaled by the angle of a half circle π, not the angle of a full circle 2π. The lower half of a circle is set symmetrically using the upper half of a circle (see Fig. 2).

4.2 Watermark insertion

The first step for watermark insertion is analyzing image contents to extract the patches, and then the watermark is inserted repeatedly into all patches. Our watermark insertion process is shown in Fig. 3.

(Step a) To extract circular patches, we apply circular Hough transform, as explained in Section 3. There are several patches in an image. We insert the watermark into all patches to increase the robustness of our method.

(Step b.1) We generate the circular-shaped watermark dependent on the size of each patch by following the method described in Section 4.1. Since we have endeavored to construct the patches such that their radius is similar to, or larger than, the x and y magnitudes of the rectangular watermark during extracting circular patches, a pixel of the rectangular watermark w is mapped to several pixels in the circular-shaped watermark wc. This compensates for errors in alignment of the circular patches regarding location and scale during watermark detection.

(Step b.2) The insertion of the watermark must not affect the perceptual quality of images. This constraint refines the insertion strength of the watermark. Human visual system should be considered to insert imperceptibly. We apply the perceptual mask of (3) [14].

$$\Lambda = \alpha \cdot (1 - \mathrm{nvf}) + \beta \cdot \mathrm{nvf} \tag{3}$$

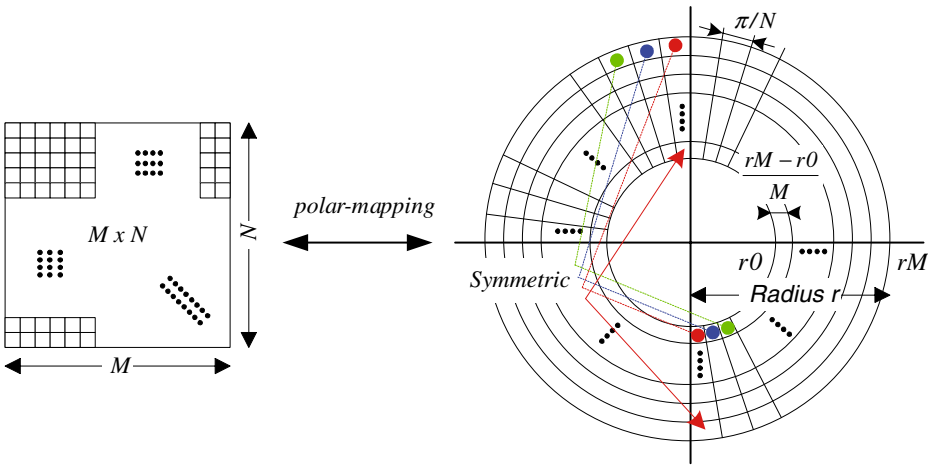


Fig. 2 Polar-mapping between the *rectangular* watermark and the *circular shaped* watermark

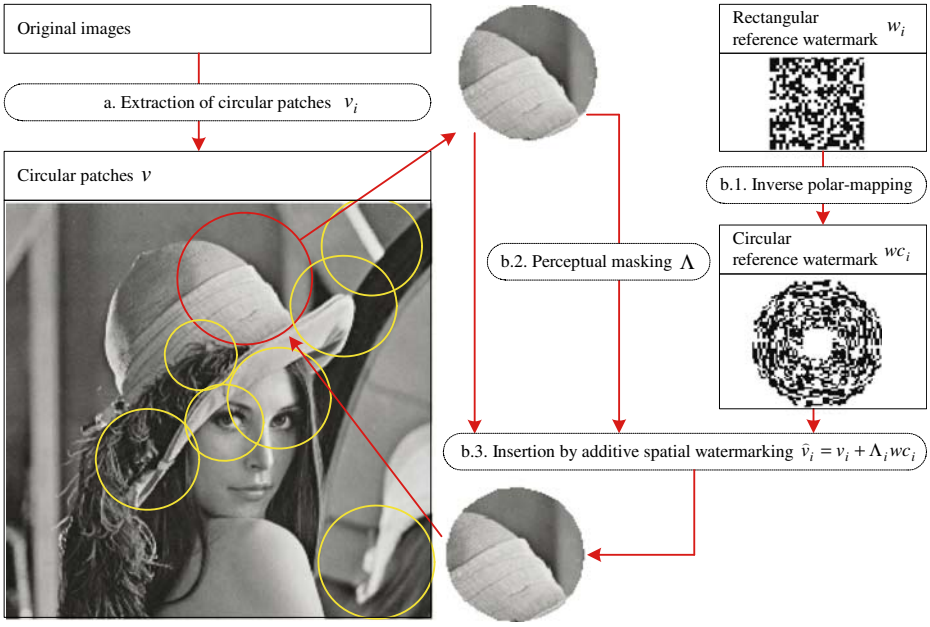


Fig. 3 Watermark insertion process

where α is the lower bound of visibility in flat and smooth regions and β is the upper bound in edged and textured regions. The noise visibility function is calculated as following (4):

$$\mathrm{NVF}(i,j) = \frac{1}{1+\theta\cdot\sigma_x^2(i,j)},\ \theta = \frac{D}{\sigma_{x_{\max}}^2} \tag{4}$$

where $\sigma_x^2(i,j)$ and $\sigma_{x_{\max}}^2$ denote local variance and its maximum, respectively. D is a scaling constant.

(Step b.3) Finally, we insert this circular-shaped watermark in an additive way in the spatial domain. The insertion of the watermark is represented as the spatial addition between the pixels of the image and the pixels of the circular-shaped watermark as (5):

$$\widehat{v_i} = v_i + \Lambda_i \mathrm{wc}_i \text{ where } \mathrm{wc}_i \approx N(0,1) \tag{5}$$

v_i and wc_i denote the pixels of the image and the circular-shaped watermark, respectively. Λ represents the perceptual mask to control the insertion strength of the watermark.

4.3 Watermark detection

Similarly to watermark insertion, the first step for watermark detection is analyzing contents to find patches. The watermark is then detected from the patches. If the watermark is

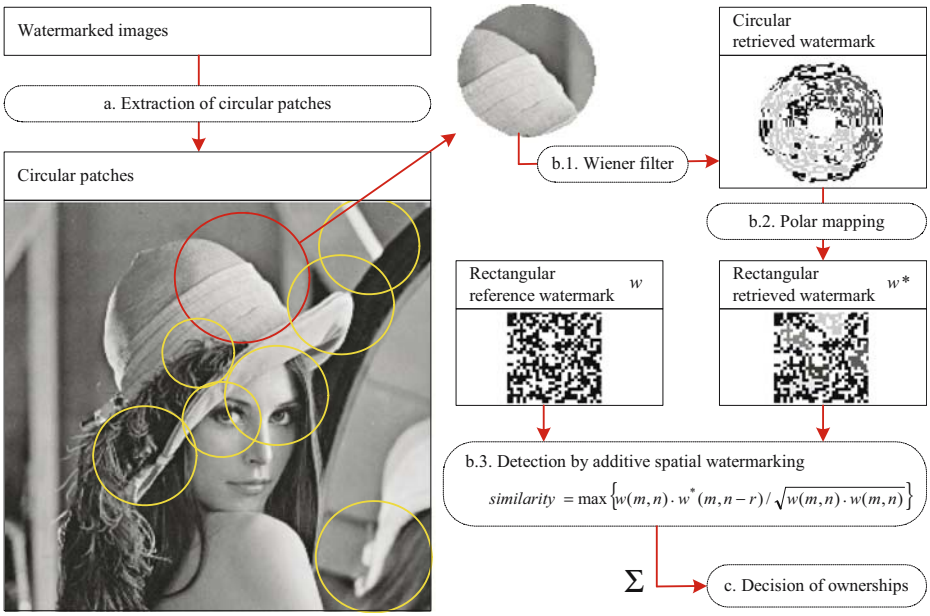


Fig. 4 Watermark detection process

correctly detected from more than one patch, we can prove ownership successfully. Our process for watermark detection is shown in Fig. 4.

(Step a) To extract circular patches, we use circular Hough transform, as described in Section 3. There are several patches in an image and we need to try to detect the watermark from all patches.

(Step b.1) The additive watermarking method in the spatial domain inserts the watermark into image contents as noise. We apply a Wiener filter to calculate this noise by calculating difference between the watermarked image and its Wiener filtered image and then regard it as the retrieved watermark. As with the watermark insertion process, we compensate for the modification by the perceptual masks, but such compensation does not have a great affect the performance of watermark detection.

(Step b.2) To measure similarity between the reference watermark generated during watermark insertion and the retrieved watermark, the retrieved circular watermark should be converted into the rectangular watermark by applying the polar mapping explained in Section 4.1. Considering the fact that the watermark is inserted symmetrically, i.e. that the upper half of a circle is symmetric to the lower half of a circle, we take the mean value from two regions. As mentioned in Section 4.1, by this mapping, the watermark can be detected using the correlation detector.

(Step b.3) We apply circular convolution to the reference watermark and the retrieved watermark. The degree of similarity between the two, called the response of

the watermark detector, is represented by the maximum value of circular convolution as (6):

$$\text{similarity} = \max\left\{\frac{w(m,n)\cdot w^{*}(m,n-r)}{\sqrt{w(m,n)\cdot w(m,n)}}, \text{for } r=[0,n]\right\} \tag{6}$$

where w is the reference watermark and w* is the retrieved watermark. The range of similarity values is from −1 to 1. We can identify the rotation angle (π/r) of the patches by finding the r with the maximum value. If the similarity exceeds a pre-defined threshold, we can be satisfied that the reference watermark has been inserted. The way to determine the threshold will be described in the following section.

(Step c) As mentioned before, there are several circular patches in an image. Therefore, if the watermark is detected from at least one patch, we can demonstrate ownership successfully. When the watermark is not detected from any patch, our watermarking method fails. However, it is highly likely that the proposed method will detect the watermark after attacks because we insert the watermark into several circular patches, not just one.

Our watermarking scheme is robust against geometric distortion attacks as well as signal processing attacks. Scaling and translation invariance is achieved by extracting circular patches from the circular Hough transform. Rotation invariance is obtained successfully by using the translation property of the polar-mapped circular patches.

4.4 Error probability analysis

Since ownership is verified by deciding whether or not the similarity exceeds a predefined threshold, the probability that our watermark detector will generate errors depends on selection of the threshold. We should consider false positive error and false negative error. False positive error is the probability that the watermark will be detected correctly when images are not watermarked. False negative error is the probability that the method will fail to retrieve the inserted watermark from watermarked images. In practice, it is difficult to analyze false negative error because of various attacks. It is common to select the threshold based on false positive error.

In order to estimate the error probability of our watermark detector, we attempted to retrieve 20 random watermarks in 100 randomly collected images. 30,120 circular patches were processed to detect the watermarks, because each image contained several patches. The size of the rectangular watermark was 32×32 pixels. The histogram of similarity values (normalized correlation) and its gamma distribution are shown in Fig. 5.

In most cases, the simplest way to estimate the error probability is to assume that the distribution of detection values follows a Gaussian distribution model [4]. However, the distribution of the response of our watermark detector follows more of a Gamma distribution model, since we take the maximum value from circular convolution. Gamma distribution as defined as (7):

$$f(x;\alpha,\beta) = \begin{cases} \frac{1}{\beta^{\alpha}\Gamma(\alpha)} x^{\alpha-1} e^{-x/\beta} & x \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

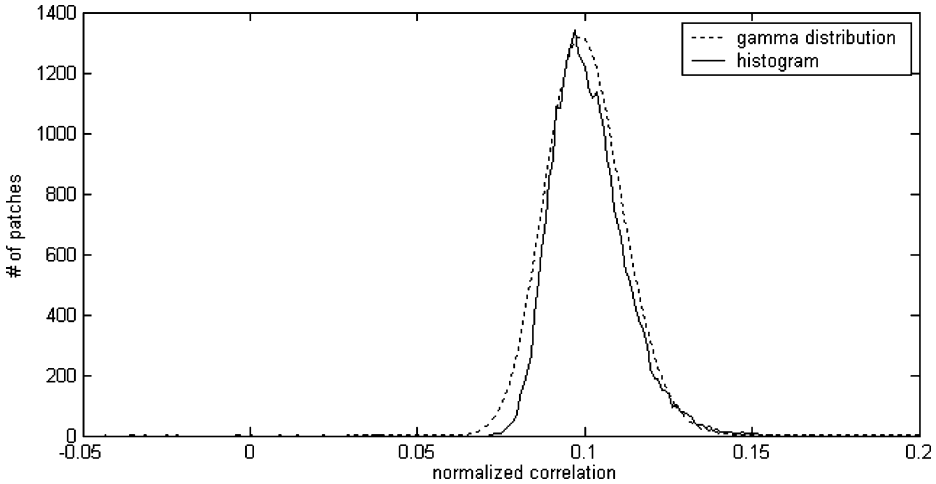


Fig. 5 Histogram of similarity values and its Gamma distribution

where x is a continuous random variable. Parameters α and β satisfy $\alpha>0$ and $\beta>0$ and are calculated using the mean and variance of random variable x as (8):

$$E(x) = \mu_x = \alpha\beta, V(x) = \sigma_x^2 = \alpha\beta^2 \tag{8}$$

Based on the results, the mean and variance of the detector response x were 0.1011 and 1.3359e–004, and the values of parameters α and β were 76.5439 and 0.0013, respectively. Table 1 shows the error probability of our watermark detector generating a false positive error and its threshold.

5 Experiment results

We carried out experiments to measure (1) the robustness of circular Hough transform to image distortions and (2) the performance of the proposed watermarking method. We applied most of the attacks listed in Stirmark 3.1 to 10 images that were commonly used in image-processing applications (all images were 512×512 pixels; see Fig. 6).

5.1 Performance of circular Hough transform

To analyze the robustness of circular Hough transform, we first extracted circular patches from both the original image and attacked images, and then compared the location and

Table 1 Error probability of our watermark detector and its threshold

Error probability	Thresholds
10^{-6}	0.1659
10^{-7}	0.1731
10^{-8}	0.1799
10^{-9}	0.1865

Fig. 6 Test images: Lena, Baboon, Pepper, Plane, Lake, Boat, Bridge, Indian, Pentagon, and Couple

radius of the patches from the original image with those of the patches from the attacked images. Prior to the comparison, we reversed the location and radius of circular patches from geometric attacked images to those of patches in the original image. If the difference between circular patches from the original image and those from attacked images was below 2 pixels, we regarded the patches as having been re-detected correctly.

The number of extracted circular patches was 9, 10, 14, 10, 14, 12, 13, 10, 13, and 10 in the Lena, Baboon, Pepper, Plane, Lake, Boat, Bridge, Indian, Pentagon, and Couple images, respectively. The more textured an image is, the more patches it has. The number of extracted circular features depends on the image content. If the image includes many objects that may be approximated to circles, for example Pepper, Lake, and Pentagon, we can extract many circular features, but if the image is composed of simple objects or has complex textures, for example Lena, Baboon, Plane, and Couple, it will be difficult to acquire a sufficient number of circular features for robust watermarking. In our method, prior to extracting circular features, we applied a Gaussian filter to make the curvature of objects circular and increase the robustness of circular Hough transform. Moreover, circular features could be distributed over the whole image.

We applied signal processing attacks (median filter, Gaussian filter, additive uniform noise, and JPEG compression) and geometric distortion attacks (cropping, rotation, and scaling). Table 2 shows the number of circular patches redetected correctly under each image attack. As expected, most of the circular patches from the original image were re-detected correctly under image attacks. We extracted most of the circular patches with signal processing attacks and a considerable number of the circular patches even after geometric distortion attacks. In geometric distortion attacks, the cropped areas increased in proportion to the strength of attacks, so that the detection ratio fell. These results support the contention that circular Hough transform is robust to image attacks and useful for robust watermarking.

5.2 Performance of the proposed watermarking method

We tested the performance of the proposed method. The size of the watermark was 32×32 pixels and the weighting factors α and β of the noise visibility function were set 5.0 and 1.0, respectively. We achieved a 10^{-8} error probability (reliability) for the proposed method by setting the threshold at 0.18 and tried to retrieve the inserted watermark by searching the radius of circular features in the range −2 to +2 to compensate for misalignment errors of circular Hough transform.

Table 2 Number of correctly re-detected circular patches following image attacks

	Lena	Baboon	Pepper	Plane	Lake	Boat	Bridge	Indian	Penta.	Couple
Median 2×2	9 / 9	10 / 10	14 / 14	10 / 10	14 / 14	11 / 12	10 / 13	10 / 10	13 / 13	8 / 10
Median 3×3	9 / 9	9 / 10	13 / 14	10 / 10	14 / 14	9 / 12	11 / 13	10 / 10	13 / 13	8 / 10
Median 4×4	9 / 9	9 / 10	13 / 14	10 / 10	12 / 14	9 / 12	11 / 13	10 / 10	11 / 13	9 / 10
Gaussian filter	9 / 9	10 / 10	13 / 14	10 / 10	14 / 14	11 / 12	13 / 13	10 / 10	12 / 13	10 / 10
Uniform noise	8 / 9	9 / 10	12 / 14	10 / 10	13 / 14	11 / 12	11 / 13	10 / 10	11 / 13	8 / 10
JPEG compress. 40	9 / 9	10 / 10	14 / 14	9 / 10	13 / 14	11 / 12	13 / 13	10 / 10	11 / 13	9 / 10
JPEG compress. 50	8 / 9	10 / 10	14 / 14	10 / 10	14 / 14	11 / 12	13 / 13	10 / 10	12 / 13	9 / 10
JPEG compress. 60	9 / 9	10 / 10	14 / 14	10 / 10	14 / 14	11 / 12	13 / 13	9 / 10	12 / 13	9 / 10
JPEG compress. 70	9 / 9	9 / 10	14 / 14	10 / 10	14 / 14	11 / 12	13 / 13	10 / 10	12 / 13	9 / 10
JPEG compress. 80	9 / 9	10 / 10	14 / 14	10 / 10	14 / 14	12 / 12	13 / 13	10 / 10	12 / 13	10 / 10
JPEG compress. 90	9 / 9	9 / 10	14 / 14	10 / 10	14 / 14	11 / 12	13 / 13	10 / 10	12 / 13	10 / 10
Crop 5%	8 / 9	8 / 10	10 / 14	10 / 10	12 / 14	7 / 12	11 / 13	10 / 10	8 / 13	9 / 10
Crop 10%	7 / 9	7 / 10	8 / 14	9 / 10	9 / 14	5 / 12	10 / 13	9 / 10	6 / 13	7 / 10
Crop 15%	6 / 9	7 / 10	7 / 14	5 / 10	8 / 14	4 / 12	9 / 13	7 / 10	6 / 13	7 / 10
Crop 20%	6 / 9	5 / 10	6 / 14	6 / 10	7 / 14	4 / 12	7 / 13	6 / 10	4 / 13	7 / 10
Crop 25%	5 / 9	5 / 10	5 / 14	5 / 10	5 / 14	3 / 12	6 / 13	4 / 10	6 / 13	6 / 10
Crop 50%	4 / 9	2 / 10	3 / 14	2 / 10	2 / 14	2 / 12	3 / 13	1 / 10	1 / 13	3 / 10
Rotation 0.5°+Crop	9 / 9	10 / 10	12 / 14	10 / 10	14 / 14	11 / 12	13 / 13	10 / 10	12 / 13	8 / 10
Rotation 1.0°+Crop	8 / 9	9 / 10	12 / 14	10 / 10	13 / 14	11 / 12	11 / 13	10 / 10	11 / 13	9 / 10
Rotation 2.0°+Crop	8 / 9	9 / 10	11 / 14	10 / 10	13 / 14	9 / 12	11 / 13	10 / 10	11 / 13	7 / 10
Rotation 5.0°+Crop	8 / 9	9 / 10	11 / 14	10 / 10	13 / 14	10 / 12	11 / 13	10 / 10	9 / 13	9 / 10
Rotation 10.0°+Crop	8 / 9	8 / 10	9 / 14	9 / 10	11 / 14	7 / 12	10 / 13	10 / 10	9 / 13	10 / 10
Rotation 15.0°+Crop	7 / 9	8 / 10	9 / 14	8 / 10	11 / 14	6 / 12	10 / 13	9 / 10	9 / 13	8 / 10
Rotation 30.0°+Crop	6 / 9	8 / 10	7 / 14	9 / 10	10 / 14	6 / 12	10 / 13	9 / 10	8 / 13	8 / 10
Rotation 45.0°+Crop	5 / 9	8 / 10	6 / 14	8 / 10	12 / 14	7 / 12	9 / 13	8 / 10	10 / 13	9 / 10
Scaling 0.8×	3 / 9	6 / 10	8 / 14	6 / 10	6 / 14	3 / 12	4 / 13	6 / 10	6 / 13	5 / 10
Scaling 0.9×	7 / 9	5 / 10	9 / 14	7 / 10	10 / 14	7 / 12	7 / 13	8 / 10	7 / 13	6 / 10
Scaling 1.1×+Crop	8 / 9	7 / 10	10 / 14	6 / 10	13 / 14	9 / 12	10 / 13	10 / 10	11 / 13	9 / 10
Scaling 1.2×+Crop	5 / 9	4 / 10	5 / 14	5 / 10	7 / 14	6 / 12	8 / 13	8 / 10	5 / 13	8 / 10
Scaling 1.3×+Crop	4 / 9	4 / 10	3 / 14	3 / 10	4 / 14	4 / 12	2 / 13	5 / 10	4 / 13	6 / 10
Scaling 1.4×+Crop	1 / 9	2 / 10	1 / 14	2 / 10	3 / 14	2 / 12	1 / 13	4 / 10	3 / 13	4 / 10

The number of watermarked circular patches was 9, 10, 14, 10, 14, 12, 13, 10, 13, and 10 in the Lena, Baboon, Pepper, Plane, Lake, Boat, Bridge, Indian, Pentagon, and Couple images, respectively. These numbers are equal to those for the circular patches from Section 5.1. We applied signal processing attacks (median filter, Gaussian filter, additive uniform noise, and JPEG compression) and geometric distortion attacks (cropping, linear geometric transform, random bending, row–column removal, shearing, rotation + cropping, scaling + cropping, and rotation + scaling + cropping) in Stirmark 3.1.

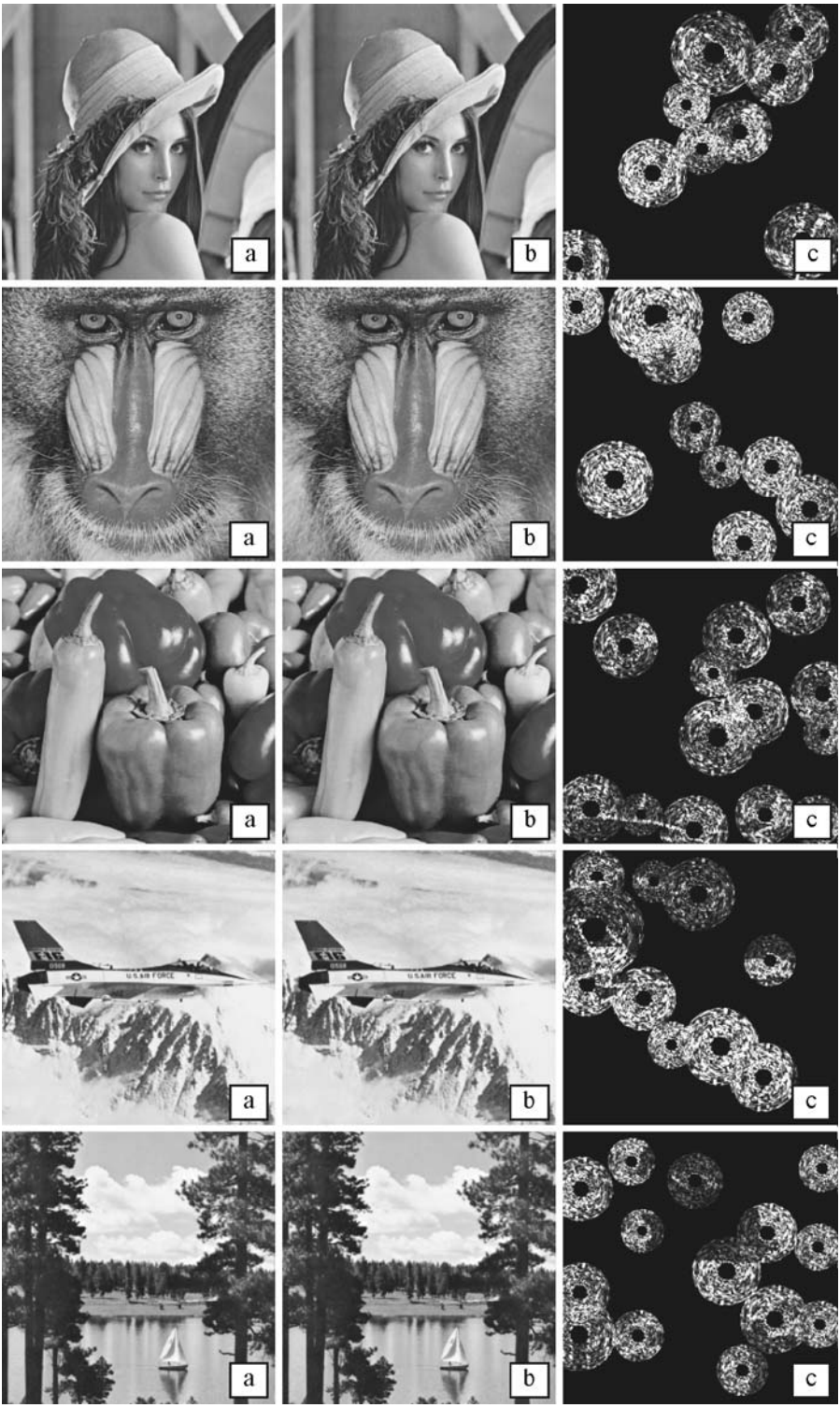
The PSNR value between the original image and the watermarked image is summarized in Table 3. We inserted the watermark into circular patches and images were modified only

Table 3 Peak Signal to Noise Ratio (PSNR)

	Lena	Baboon	Pepper	Plane	Lake	Boat	Bridge	Indian	Penta.	Couple
PSNR	43.17	40.28	42.87	42.25	41.78	42.24	38.84	42.77	38.74	42.24

Table 4 Number of circular patches where the watermark is retrieved correctly following attacks

	Lena	Baboon	Pepper	Plane	Lake	Boat	Bridge	Indian	Penta.	Couple
No attack	9 / 9	7 / 10	14 / 14	10 / 10	14 / 14	11 / 12	13 / 13	10 / 10	13 / 13	8 / 10
Median 2×2	8 / 9	7 / 10	14 / 14	10 / 10	14 / 14	9 / 12	6 / 13	10 / 10	11 / 13	6 / 10
Median 3×3	9 / 9	7 / 10	13 / 14	10 / 10	14 / 14	8 / 12	10 / 13	10 / 10	13 / 13	7 / 10
Median 4×4	8 / 9	7 / 10	11 / 14	10 / 10	12 / 14	8 / 12	8 / 13	10 / 10	11 / 13	8 / 10
Gaussian filter	8 / 9	8 / 10	12 / 14	10 / 10	14 / 14	8 / 12	11 / 13	10 / 10	12 / 13	5 / 10
Uniform noise	7 / 9	8 / 10	8 / 14	7 / 10	12 / 14	8 / 12	10 / 13	8 / 10	9 / 13	5 / 10
JPEG compress. 40	8 / 9	5 / 10	9 / 14	7 / 10	8 / 14	6 / 12	11 / 13	8 / 10	11 / 13	4 / 10
JPEG compress. 50	8 / 9	6 / 10	12 / 14	7 / 10	11 / 14	7 / 12	11 / 13	8 / 10	12 / 13	5 / 10
JPEG compress. 60	8 / 9	6 / 10	13 / 14	7 / 10	13 / 14	9 / 12	11 / 13	9 / 10	12 / 13	6 / 10
JPEG compress. 70	8 / 9	7 / 10	13 / 14	9 / 10	14 / 14	10 / 12	12 / 13	10 / 10	10 / 13	5 / 10
JPEG compress. 80	8 / 9	7 / 10	13 / 14	10 / 10	14 / 14	11 / 12	12 / 13	10 / 10	11 / 13	7 / 10
JPEG compress. 90	9 / 9	7 / 10	13 / 14	8 / 10	14 / 14	11 / 12	13 / 13	10 / 10	11 / 13	7 / 10
Crop 5%	7 / 9	8 / 10	9 / 14	8 / 10	12 / 14	4 / 12	9 / 13	10 / 10	5 / 13	7 / 10
Crop 10%	7 / 9	6 / 10	7 / 14	8 / 10	9 / 14	4 / 12	10 / 13	8 / 10	5 / 13	6 / 10
Crop 15%	6 / 9	5 / 10	7 / 14	5 / 10	8 / 14	3 / 12	8 / 13	6 / 10	5 / 13	5 / 10
Crop 20%	6 / 9	5 / 10	5 / 14	6 / 10	7 / 14	3 / 12	5 / 13	5 / 10	3 / 13	6 / 10
Crop 25%	5 / 9	5 / 10	5 / 14	5 / 10	5 / 14	2 / 12	4 / 13	4 / 10	5 / 13	5 / 10
Crop 50%	3 / 9	2 / 10	2 / 14	1 / 10	1 / 14	1 / 12	3 / 13	1 / 10	1 / 13	2 / 10
Linear trans. 1.008	8 / 9	6 / 10	10 / 14	9 / 10	13 / 14	7 / 12	10 / 13	10 / 10	8 / 13	8 / 10
Linear trans. 1.011	7 / 9	7 / 10	10 / 14	9 / 10	13 / 14	8 / 12	9 / 13	10 / 10	8 / 13	8 / 10
Linear trans. 1.012	7 / 9	7 / 10	9 / 14	9 / 10	13 / 14	7 / 12	10 / 13	10 / 10	6 / 13	8 / 10
Random bending	8 / 9	4 / 10	14 / 14	10 / 10	12 / 14	8 / 12	8 / 13	10 / 10	8 / 13	7 / 10
Row/Col Removal 1 1	8 / 9	7 / 10	13 / 14	9 / 10	14 / 14	11 / 12	12 / 13	9 / 10	12 / 13	8 / 10
Row/Col Removal 1 5	8 / 9	7 / 10	12 / 14	9 / 10	11 / 14	10 / 12	12 / 13	10 / 10	12 / 13	7 / 10
Row/Col Removal 5 1	8 / 9	6 / 10	12 / 14	9 / 10	14 / 14	11 / 12	8 / 13	8 / 10	9 / 13	8 / 10
Row/Col Removal 17 5	7 / 9	4 / 10	10 / 14	8 / 10	11 / 14	7 / 12	5 / 13	9 / 10	8 / 13	7 / 10
Row/Col Removal 5 17	8 / 9	3 / 10	10 / 14	10 / 10	10 / 14	6 / 12	6 / 13	9 / 10	8 / 13	6 / 10
Shearing x 0 y 1	7 / 9	7 / 10	13 / 14	8 / 10	13 / 14	8 / 12	11 / 13	10 / 10	10 / 13	9 / 10
Shearing x 0 y 5	7 / 9	1 / 10	12 / 14	5 / 10	12 / 14	4 / 12	8 / 13	8 / 10	9 / 13	7 / 10
Shearing x 1 y 0	8 / 9	7 / 10	13 / 14	9 / 10	13 / 14	11 / 12	11 / 13	10 / 10	8 / 13	8 / 10
Shearing x 5 y 0	6 / 9	2 / 10	12 / 14	8 / 10	7 / 14	7 / 12	4 / 13	8 / 10	9 / 13	5 / 10
Shearing x 1 y 1	7 / 9	6 / 10	10 / 14	9 / 10	11 / 14	8 / 12	11 / 13	10 / 10	8 / 13	6 / 10
Rotation 0.5°+Crop	8 / 9	7 / 10	12 / 14	10 / 10	14 / 14	9 / 12	11 / 13	9 / 10	9 / 13	7 / 10
Rotation 1.0°+Crop	7 / 9	6 / 10	12 / 14	9 / 10	12 / 14	11 / 12	10 / 13	9 / 10	9 / 13	7 / 10
Rotation 2.0°+Crop	7 / 9	4 / 10	11 / 14	9 / 10	13 / 14	8 / 12	6 / 13	9 / 10	10 / 13	5 / 10
Rotation 5.0°+Crop	7 / 9	8 / 10	10 / 14	9 / 10	13 / 14	8 / 12	8 / 13	10 / 10	7 / 13	7 / 10
Rotation 10.0°+Crop	8 / 9	7 / 10	8 / 14	9 / 10	10 / 14	6 / 12	7 / 13	9 / 10	9 / 13	7 / 10
Rotation 15.0°+Crop	7 / 9	5 / 10	9 / 14	7 / 10	11 / 14	6 / 12	6 / 13	9 / 10	9 / 13	6 / 10
Rotation 30.0°+Crop	6 / 9	4 / 10	7 / 14	6 / 10	10 / 14	6 / 12	8 / 13	8 / 10	6 / 13	5 / 10
Rotation 45.0°+Crop	4 / 9	4 / 10	6 / 14	7 / 10	11 / 14	7 / 12	8 / 13	7 / 10	9 / 13	7 / 10
Scaling 0.8×	2 / 9	3 / 10	8 / 14	5 / 10	4 / 14	1 / 12	3 / 13	5 / 10	4 / 13	5 / 10
Scaling 0.9×	7 / 9	5 / 10	9 / 14	7 / 10	10 / 14	7 / 12	7 / 13	8 / 10	7 / 13	6 / 10
Scaling 1.1×+Crop	7 / 9	5 / 10	10 / 14	5 / 10	11 / 14	9 / 12	6 / 13	10 / 10	8 / 13	8 / 10
Scaling 1.2×+Crop	4 / 9	4 / 10	5 / 14	4 / 10	7 / 14	6 / 12	3 / 13	8 / 10	5 / 13	8 / 10
Scaling 1.3×+Crop	4 / 9	2 / 10	2 / 14	2 / 10	4 / 14	4 / 12	2 / 13	5 / 10	4 / 13	5 / 10
Scaling 1.4×+Crop	1 / 9	1 / 10	1 / 14	1 / 10	3 / 14	2 / 12	1 / 13	1 / 10	1 / 13	2 / 10
Rot. 0.5°+Sca.+Crop	9 / 9	7 / 10	13 / 14	10 / 10	14 / 14	12 / 12	10 / 13	10 / 10	12 / 13	11 / 10
Rot. 1.0°+Sca.+Crop	9 / 9	6 / 10	13 / 14	8 / 10	14 / 14	12 / 12	10 / 13	10 / 10	12 / 13	9 / 10
Rot. 2.0°+Sca.+Crop	8 / 9	5 / 10	12 / 14	9 / 10	13 / 14	9 / 12	11 / 13	10 / 10	11 / 13	7 / 10
Rot. 5.0°+Sca.+Crop	8 / 9	6 / 10	9 / 14	8 / 10	12 / 14	8 / 12	10 / 13	10 / 10	10 / 13	8 / 10
Rot. 10.0°+Sca.+Crop	6 / 9	4 / 10	4 / 14	7 / 10	8 / 14	7 / 12	4 / 13	6 / 10	5 / 13	8 / 10
Rot. 15.0°+Sca.+Crop	3 / 9	2 / 10	3 / 14	4 / 10	4 / 14	1 / 12	2 / 13	7 / 10	5 / 13	5 / 10


a
b
c
a
b
c
a
b
c
a
b
c
a
b
c

in parts. As a result, our method achieved high PSNR values. In the highly textured images, such as Baboon, Bridge, and Pentagon, the PSNR value was relatively low because we inserted the watermark strongly due to the fact that noise was imperceptible.

Table 4 shows the number of circular patches where the watermark was correctly retrieved, i.e. whose similarity exceeded the threshold. Our proposed method was able to detect the inserted watermark from a considerable number of circular patches even following signal processing attacks and geometric distortion attacks. If the watermark is detected from more than one circular patch, it may be said that ownership has been determined successfully.

When we transform a rectangular watermark into a circular-shaped watermark, a pixel of the rectangular watermark is mapped to a homocentric region of a circle and that compensated for small misalignment errors of circular patches. Consequently, in testing, the watermark was able to survive even linear geometric transform, random bending, row–column removal, and shearing attacks. In rotation + cropping, scaling + cropping, and rotation + scaling + cropping attacks, the cropped areas increased in proportion to the strength of attacks and hence the detection ratio fell.

The difference of matching items in Tables 2 and 4 represents the number of circular patches where the circular patch is well-synchronized but the additive watermarking method fails to detect the inserted watermark. As mentioned before, circular Hough transform can extract patches even after image attacks and in complex textured images. However, it is unlikely that the additive watermarking method can detect the watermark correctly when images contain complex texture or are distorted by noise.

Similarly to other watermarking techniques in the spatial domain, our proposed method shows relatively low performance after scale distortion. In general, image scaling requires interpolation, which distorts or attenuates the inserted watermark severely without degradation of perceptual quality. However, we inserted the watermark into an image multiple times so that our method could determine the ownership of the contents successfully.

6 Discussion and conclusions

Figure 7 shows original images, watermarked images, and residual images between the original and watermarked images. We modified the histogram of residual images for convenience. As may be seen from Fig. 7, we inserted the watermark into the image so as not to be visible to the naked eye. Residual images show the location and radius of circular patches, how the rectangular watermark is shaped to the homocentric circle, and how the additive watermarking method inserts the watermark in the spatial domain. Our method has high PSNR values because images are modified partly. When the image is well-textured and contains a number of objects approximated to circles, the patches are scattered all over the image, which allows the watermark to be inserted over the whole image. However, if the texture of the image is simple, for example the sky areas in Lake and Plane, the watermark is concentrated on the area near to the object.

Drawbacks of the proposed method are principally related to its weakness to large distortions of the aspect ratio. In addition, to the computation time required for circular

Fig. 7 **a** Original image, **b** watermarked image, and **c** residual image in Lena, Baboon, Pepper, Plane, and Lake images

Hough transform, our method cannot be used effectively in real-time applications [8]. Future work will focus on eliminating these drawbacks. More research will be also required to solve problems that arise from interpolation.

We proposed a geometrically invariant watermarking method that synchronizes the location for watermark insertion and detection by using media contents. We employed circular Hough transform to extract circular patches and these patches were watermarked additively on the spatial domain. Since circular Hough transform extracts robust features even after image attacks and the additive watermarking method detects inserted watermarks invariantly to rotation distortion, the proposed method could be resilient to geometric distortion attacks as well as signal processing attacks. Through experiments, we showed the robustness of the proposed method by applying most of the attacks listed in Stirmark 3.1. We believe that the robustness of features is important in designing robust watermarking and circular Hough transform may be a solution to the problem of extracting robust features.

Acknowledgement This work is financially supported by the Ministry of Education and Human Resources Development (MOE), the Ministry of Commerce, Industry and Energy (MOCIE) and the Ministry of Labor (MOLAB) through the fostering project of the Lab of Excellency, also partially supported by the Korea Science and Engineering Foundation (KOSEF) through the Advanced Information Technology Research Center (AITrc).

References

1. Arghoniemy M, Twefik AH (2004) Geometric invariance in image watermarking. IEEE Trans Image Process 13:145–153
2. Bas P, Chassery J-M, Macq B (2002) Geometrically invariant watermarking using feature points. IEEE Trans Image Process 11(9):1014–1028
3. Cox IJ, Kilian J, Shamoon T (1997) Secure spread spectrum watermarking for multimedia. IEEE Trans Image Process 6(12):1673–1678
4. Cox IJ, Miller ML, Bloom JA (2002) Digital watermarking, chap. 5. Morgan Kaufmann, San Francisco, CA
5. Kutter M (1998) Watermarking resisting to translation, rotation and scaling. Proc SPIE 3528:423–431
6. Kutter M, Bhattacharjee SK, Ebrahimi T (1999) Toward second generation watermarking schemes. In: Proc. of International Conference on Image Processing, vol. 1, pp 320–323
7. Lin C-Y, Cox IJ (2001) Rotation, scale and translation resilient watermarking for images. IEEE Trans Image Process 10:767–782
8. Lin ET, Delp EJ (2004) Spatial synchronization using watermark key structure. Proc SPIE 5306: 536–547
9. Nikolaidis A, Pitas I (2001) Region-based image watermarking. IEEE Trans Image Process 10(11): 1726–1740
10. Pereira S, Pun T (2000) Robust template matching for affine resistant image watermark. IEEE Trans Image Process 9:1123–1129
11. O'Ruanaidh JJK, Pun T (1998) Rotation, scale and translation invariant spread spectrum digital image watermarking. Signal Process 66:303–317
12. Simitopoulos D, Koutsonanos DE, Strintzis MG (2003) Robust image watermarking based on generalized radon transformation. IEEE Trans Circuits Syst Video Technol 13:732–745
13. Tang C-W, Hang H-M (2003) A feature-based robust digital image watermarking scheme. IEEE Trans Signal Process 51(4):950–959
14. Voloshynovskiy S, Herrigel A, Baumgartner N, Pun T (1999) A stochastic approach to content adaptive digital image watermarking. In: Proc. of International Workshop on Information Hiding, pp 212–236

Hae-Yeoun Lee received the BS degree in information engineering from Sung Kyun Kwan University, Seoul, Korea, in 1993, and MS and PhD degrees in computer science from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 1997 and 2006, respectively. From 1997 to 2001, he was with the Satellite Technology Research Center (SaTReC), KAIST, as a research student. From 2001 to 2006, he was with Satrec initiative, a company in Korea, as a senior researcher. Currently, he is a post-doctoral researcher in Weill Medical College of Cornell University, United States. His research interests include image processing, computer vision, multimedia processing, remote sensing, digital watermarking, and digital rights management.

Choong-hoon Lee received his BE degree in computer engineering in 1996 from Dongguk University, Seoul, Korea, and his MS and PhD degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1998 and 2004 respectively. From 2004 to 2005, he was a post-doctoral researcher in computer science division in KAIST where he was an invited professor from 2005 to 2006. Since 2006, he has been a senior researcher in DRM Lab in DM R&D Center, Samsung Electronics co., LTD. His major interests are digital watermarking, fingerprinting, DRM, and image/video coding.

Heung-Kyu Lee received the BS degree in electronics engineering from the Seoul National University, Seoul, Korea, in 1978, and MS, PhD Degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), in 1981, and 1984, respectively. Since 1986 he has been a professor of the Department of Computer Science, KAIST, Korea. He is an author/coauthor of more than 100 international journal and conference papers. He has been a reviewer of many international journals, J. of Electronic Imaging, Real-Time Imaging, IEEE Trans. on Circuits and Systems for Video Technology, etc. He was also program chairman of many international conferences including the Int. Workshop on Digital Watermarking (IWDW) in 2004, IEEE Int. conf. on Real-Time Systems and Applications, etc. He is now a general director of Center of Fusion Technology for Security (CFTS). His major interests are digital watermarking/ fingerprinting, media forensics, and steganography.