Available online at www.sciencedirect.com

ScienceDirect

Signal Processing 87 (2007) 1264–1280

SIGNAL PROCESSING

www.elsevier.com/locate/sigpro

# A robust content-based digital image watermarking scheme

Xiaojun Qi*, Ji Qi

*Department of Computer Science, Utah State University, Logan, UT 84322-4205, USA*

Received 5 January 2006; received in revised form 1 October 2006; accepted 2 November 2006
Available online 30 November 2006

**Abstract**

This paper presents a content-based digital image-watermarking scheme, which is robust against a variety of common image-processing attacks and geometric distortions. The image content is represented by important feature points obtained by our image-texture-based adaptive Harris corner detector. These important feature points are geometrically significant and therefore are capable of determining the possible geometric attacks with the aid of the Delaunay-tessellation-based triangle matching method. The watermark is encoded by both the error correcting codes and the spread spectrum technique to improve the detection accuracy and ensure a large measure of security against unintentional or intentional attacks. An image-content-based adaptive embedding scheme is applied in discrete Fourier transform (DFT) domain of each perceptually high textured subimage to ensure better visual quality and more robustness. The watermark detection decision is based on the number of matched bits between the recovered and embedded watermarks in embedding subimages. The experimental results demonstrate the robustness of the proposed method against any combination of the geometric distortions and various common image-processing operations such as JPEG compression, filtering, enhancement, and quantization. Our proposed system also yields a better performance as compared with some peer systems in the literature.

*Keywords:* Geometrically invariant digital watermarking; Important feature points; Image-texture-based adaptive Harris corner detector; Delaunay-tessellation-based triangle matching; Spread spectrum

## 1. Introduction

The development of high-speed Internet and compression technology allows the widespread use of multimedia applications. Nowadays, digital documents can be distributed via the World Wide Web to a large number of people in a cost-efficient way. Protection of multimedia information, especially its copyright, attracts more and more attention. There is a strong need to keep the distribution of digital multimedia works both profitable for the owner and reliable for the customer. Consequently, digital watermarking emerges as one possible and popular solution. Watermarking, also called tamper-proofing or content verification, hides a secret and personal message to protect a product's copyright or to demonstrate its data integrity. In contrast to cryptography, which immediately arouses suspicion of something secret or valuable, the watermark hides a message within digital media without noticeable changes to the host.

In watermarking applications, the robustness of the watermark to common image processing and geometric attacks is essential to the system [1].

*Corresponding author. Tel.: +1 435 797 8155; fax: +1 435 797 3265.

*E-mail addresses:* xqi@cc.usu.edu (X. Qi), jiqi@cc.usu.edu (J. Qi).

0165-1684/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2006.11.002

However, geometric manipulations are difficult to tackle because they can introduce the synchronization errors into the watermarking system and render the watermark detection impossible. A lot of research has been conducted to reduce or prevent the asynchronous problem caused by geometric distortions. These methods can be roughly divided into the following three categories:

- Template-based watermarking methods [2–4]: These methods intentionally embed additional templates into the image. These templates function as anchor points for the alignment and therefore assist the watermark synchronization in detection process.
- Invariance-domain-based watermarking methods [5–8]: These methods generally provide rotation, scaling, and translation (RST) invariant domains, namely log-polar domain [7] and Fourier–Mellin transformation domain [5,6,8], for embedding the watermark and maintaining synchronization under affine transforms.
- Moment-based watermarking methods [9–13]: These methods utilize the geometric invariants of the image including ordinary moments [9–11] and normalized Zernike moments [12,13] to reduce the synchronization errors in watermark detection process.

In general, all these aforementioned self-synchronizing watermarking schemes can somehow detect the watermark after either geometric distortions such as RST transformations or local distortions like Stirmark attacks. However, their robustness is limited. For instance, the image independent template features can be exploited to derive a new attack [14] to destroy the templates without any prior knowledge. The interpolation accuracy associated with the invariance domain and the discretization errors associated with the moments increase the synchronization errors and therefore make embedding and detection misaligned. As a result, a second-generation watermarking scheme [15] (i.e., a content-based watermarking scheme) is widely exploited to restore the synchronization of the watermark. This scheme extracts the image content in terms of the feature points and embeds the watermark into the regions associated with the image content. The extracted image feature points represent invariant references to geometric transformations and therefore can be used as anchor points for self-synchronization. A few content-based watermarking approaches are briefly reviewed here.

Bas et al. [16] use the Harris detector for feature extraction. These feature points are combined with a Delaunay tessellation to mark each triangle for embedding the watermark. The original watermark triangle will be warped during the detection to correlate with the corresponding marked triangles. Simulation results show that the robustness of the scheme depends on the capacity of the Harris detector to preserve feature points after geometric transformations, especially on images with more texture and images with less texture and large homogeneous areas. This scheme is also not robust to most common image processing attacks except JPEG compression. Tang and Hang [17] adopt the Mexican hat wavelet scale interaction method to extract feature points. They embed and extract the watermark in the normalized disks centered at the extracted feature points. However, this scheme performs well under only mild geometric distortions and certain common image processing attacks due to the interpolation errors induced by the normalization. In [15,18], the same Mexican hat wavelet method is used to locate feature points, which form the Voronoi diagrams for watermark embedding and detection. Experimental results show that both schemes are robust to high-quality JPEG compression and affine transformations. However, the watermarks have to be searched throughout the rotated images.

Several hybrid approaches [19–23] have also been employed to counterattack geometric distortions. For example, Delannay and Macq [19] embed an image-partition-based secret binary mask to modulate a spectral spreading of the synchronization mark for resisting template attacks. However, mask recovering may present an error which reduces the synchronization accuracy. Su et al. [20] apply segmentation to determine feature-based spatially localized structures for watermark embedding and detection. This method offers good tolerance to collusion attacks and reasonable robustness to geometric distortions. Kang et al. [21] use a DWT–DFT composite image watermarking algorithm to provide robustness against both affine transformations and JPEG compression. However, the scheme is not robust to common image-processing attacks.

In this paper, we develop a robust content-based watermarking scheme. This scheme combines the advantages of important feature extraction, Delaunay-tessellation-based triangle matching, perceptual analysis, one-way hash functions, error correcting codes, and spread-spectrum-based blind watermark embedding and retrieval to reduce the watermark synchronization problem and resist geometric distortions and common image processing attacks. Section 2 describes the proposed texture-based adaptive image content extraction method. Section 3 briefly reviews the variants of several important techniques used in our scheme. Section 4 contains the description of our watermark embedding procedure. Section 5 covers the details of our watermark detection procedure. Section 6 shows the simulation results by comparing our scheme with two content-based approaches in terms of robustness against both geometric distortions and common image processing attacks. In addition, we also demonstrate the performance of our proposed watermarking scheme on 105 images of different textures under different Stirmark attacks. Section 7 concludes this presentation.

## 2. Image content extraction

Extracting the image content in terms of the feature points is an important step in the proposed digital image-watermarking scheme. In order to detect watermarks without access to the original images, we look for the image content that is perceptually significant and can, thus, resist various types of common image processing and geometric distortions. The image-content-bounded feature points can be further used as synchronization markers in watermark detection.

### *2.1. The common Harris corner detector*

Bas et al. [16] evaluate the performance of three commonly used detectors (i.e., the Harris corner detector [24], the Achard-Rouquet detector [25], and the SUSAN detector [26]) and conclude the Harris corner detector is the most robust. Schmid et al. [27] also compare the Harris corner detector with the Heitger detector [28], the Forstner detector [29], the Horaud detector [30], and the Cottier detector [31] with regards to the repeatability of the detector when the detected corner points are used for matching purposes. That is, for different distorted versions of the same scene, the detector should be able to extract similar, if not identical, points, despite variations due to a change of orientation or sharpness. The results of these studies prove the Harris detector is the most stable. The commonly used Harris corner detector refines the detection function [32] by using the following shape-factor-based matrix:

$$M(x,y)=\begin{bmatrix} A_{x,y} & C_{x,y} \\ C_{x,y} & B_{x,y} \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial I(x,y)}{\partial x}\right)^2 & \left(\frac{\partial I(x,y)}{\partial x}\right)\left(\frac{\partial I(x,y)}{\partial y}\right) \\ \left(\frac{\partial I(x,y)}{\partial x}\right)\left(\frac{\partial I(x,y)}{\partial y}\right) & \left(\frac{\partial I(x,y)}{\partial y}\right)^2 \end{bmatrix}, \quad (1)$$

where $I(x,y)$ is the gray level intensity, $\partial I(x,y)/\partial x \approx I(x,y)*[-1,0,1]$, $\partial I(x,y)/\partial y \approx I(x,y) *[-1,0,1]^{\mathrm{T}}$, and $*$ denotes the convolution product. The corner points are located at the positions with large corner response values, which are determined by the corner response function $R(x,y)$:

$$R(x,y)=\det(M(x,y))-k[\mathrm{trace}(M(x,y))]^2 = \left(A_{x,y}B_{x,y}-C_{x,y}^2\right)-k\left(A_{x,y}+B_{x,y}\right)^2, \quad (2)$$

where $k$ is a constant that is set to be 0.04.

### *2.2. The image-texture-based adaptive Harris corner detector*

The image content extraction algorithm uses our image-texture-based adaptive Harris corner detector to find important feature points (i.e., corner points) to reduce the synchronization errors in watermark detection. The major contributions are:

- Apply some pre-processing techniques to reduce the noise effect.
- Regulate the number of important feature points based on the texture of the image.

The following four steps detail the image content extraction procedure:

1. Apply a Gaussian low-pass filter to original image $I(x,y)$ to avoid corners due to image noise.
2. Apply a rotationally symmetric 3 × 3 Gaussian low-pass filter with the standard deviation of 0.5 to three derivative images, namely, $A_{x,y}$ $B_{x,y}$ and $C_{x,y}$, to achieve additional resistance to possible image noise.
3. Calculate $R(x,y)$ within a circular window, which is at the image center and covers the largest area of the original image. The resulting function

reduces the effect of image-center-based rotation attacks.

4. Apply a threshold $T$ on $R(x,y)$ and search for important feature points based on the local maxima

$$\{R(x,y)|R(x,y)>T \wedge R(x,y) \geq R(u,v), \forall(u,v) \in V_{x,}\} \tag{3}$$

where $T$ is a predefined threshold value that is empirically set to be $10^6$ in our scheme to extract a desired number of corner points, and $V_{x,y}$ represents a circular neighborhood centered at $(x,y)$.

We choose the circular neighborhood window to avoid the increasing detector anisotropy and to obtain a homogeneous distribution of feature points in the image. It is also important to determine the appropriate window size. If the window is too small, the distribution of feature points is concentrated on textured areas. If the window is too large, the feature points become isolated. Fig. 1 illustrates the effect of different window sizes on the resultant feature points, which are shown as large white squares for display purpose. We can easily observe that the number of detected feature points remains relatively constant to the window size of the detector. That is, decreasing the window size will increase the number of detected feature points, and vice versa. It therefore follows that one can increase the good matches of feature points on both the original and probe images by decreasing the size of the window. However, this is done at the price of a proportional increase of the total number of corner points to analyze. In order to compensate, we determine a suitable window size based on the dimension and texture of the image. The diameter of the circular window is calculated:

$$D = \sqrt{\frac{wh}{np}}, \tag{4}$$

where

- Integers $w$ and $h$, respectively, represent the width and height of the image.
- Integer $p$ is an empirical value for obtaining a reasonable number of feature points for images with large homogeneous areas. It is set to be 60 in our implementation.
- Integer $n$ is the window size quantizer, which depends on the texture of the image. It is set to be:

$$n = \begin{cases} 1.5, & \text{if ratio} \geqslant 0.01 \text{(high texture)}, \\ 2.5, & \text{if ratio} \geqslant 0.002 \text{(medium texture)}, \\ 3.5, & \text{if ratio} \geqslant 0.0001 \text{(low texture)}, \end{cases} \tag{5}$$

where the ratio is computed as the proportion of the feature points to the total number of pixels in the image. These feature points are obtained by using our proposed adaptive Harris corner detector with a $3 \times 3$ neighborhood window.

With an adaptive and optimized window size for the Harris corner detector, we can make sure a certain amount of corner points can always be detected, neither too many, nor too few. By doing this, the computational cost of the image content extraction and the Delaunay-tessellation-based triangle matching can be automatically balanced in watermark detection.

Fig. 2 demonstrates the extracted important feature points by applying our image-texture-based

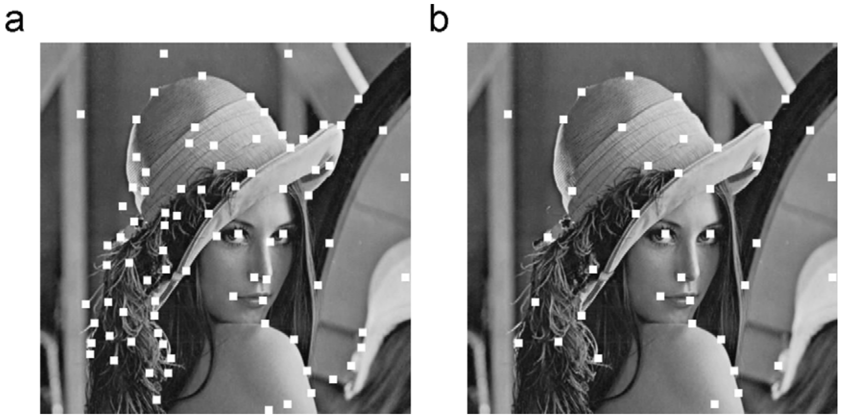

Fig. 1. The effect of different window sizes on the feature points. (a) Window size of 26 × 26 and (b) window size of 52 × 52.

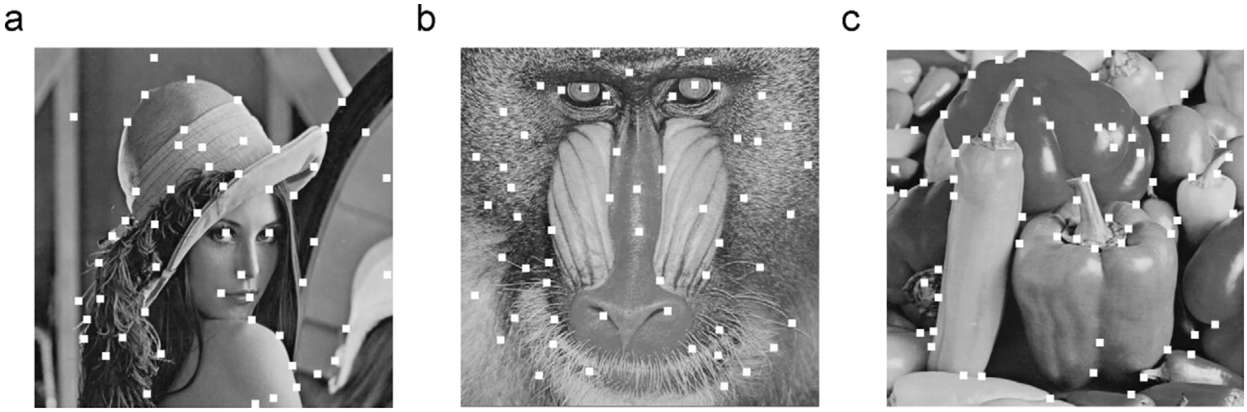

Fig. 2. Important feature points extracted by our image-texture-based adaptive Harris corner detector.

adaptive Harris corner detector on three images (e.g., Lena, Baboon, and Pepper) with different textures. For example, Lena includes large homogeneous areas with sharp edges; Baboon includes textured areas with high frequency components; and Pepper includes large homogeneous areas without sharp edges and therefore is an example of the low textured image. As shown in Fig. 2, the number of important feature points is regulated by the texture of the image. That is, medium and low textured images such as Lena and Pepper have enough image content for watermark detection process to reduce the synchronization errors. Similarly, high textured image such as Baboon does not have overwhelming number of image content as suffered by other content-based methods. On the contrary, it also possesses sufficient image content for watermark detection process to achieve self-synchronization. It is worthwhile to mention that the extremely low textured image (i.e., ratio $<0.0001$) will not be considered as a candidate image for copyright protection due to insufficient important feature points for self-synchronization.

## 3. Variants of related techniques

The variants of several important techniques employed in the proposed system are briefly introduced in this section.

### 3.1. Error correcting codes

Error correcting codes [33] are incorporated into our system to overcome the corruption of a watermark in the channeled communication since a communication channel of the watermarking procedure is not noise free and will subject to intentional or unintentional distortions. In our system, we choose Hamming codes as an error detection mechanism to correct single bit errors. Specifically, the 8-bit watermark sequence is divided into two 4-bit subsequences. The Hamming code is then applied to each subsequence to generate its (7, 4) single-bit error correcting code. Since transmitting 4-bit data always yields more errors than transmitting its (7, 4) single-bit error correcting code, error-correcting capability ensures a better quality signal at the receiver and a higher recovery rate in watermark detection. Consequently, it increases the possibility of the perfect match in any embedding subimage and reduces the false negative probability.

### 3.2. The PN-sequence and one-way hash functions

The pseudo-noise (PN) sequence based spread spectrum method [34] is used in our system. This sequence combined with the watermark is adaptively embedded into mid-frequency positions in DFT domain. A variant of the one-way hash function [35] is used in our system to generate the highly secure mid-frequency positions. The six steps for generating these positions are:

1. Save all middle frequency positions into vector $V$.
2. Randomly choose two large prime numbers $p$ and $q$, and compute the secrete key $n = pq$.
3. Obtain seed $Y$ using the encipher process:

$$X = m^K mod\, n; \quad Y = X^2 mod\, n; \tag{6}$$

where $m$ is the numerical serial number for registering the original image and $K$ is the second secret key.

4. Calculate an index $l$ by:

$$Y = Y^2 mod n; \quad l = (Y mod\, n)\ mod\ length(V); \tag{7}$$

5. Choose the $l$th item in $V$ as the embedding position and remove it from $V$ so no duplicated positions are produced.
6. Repeat steps 4–5 until the total number of embedding positions is reached.

These highly secure embedding positions can be easily reproduced by the same secret keys $n$ and $K$. In the meantime, the reproduction of these positions is computationally infeasible without knowing $n$ and $K$. To ensure the attackers cannot find out the watermark embedding positions by comparing several watermarked copies, different secret keys are used to generate the embedding positions for each embedding subimage.

### 3.3. Blind embedding and retrieval

It is always desirable to extract the watermark independent of the original image since it takes a lot of space to store the original image. The blind retrieval scheme of MPEG video watermark [36,37] is adopted in our system to achieve this goal. We employ this blind retrieval scheme in DFT instead of DCT domain. In addition, the multiplicative instead of additive embedding is applied in our modified scheme.

## 4. Watermark embedding scheme

Our watermark is designed for copyright protection. We view all possible embedding subimages as independent communication channels. To improve the robustness of the transmitted watermark bits, all channels carry the same copy of the chosen watermark. During the detection process, we claim the existence of watermark if one copy of the embedded watermark is correctly detected in one embedding subimage.

The watermark embedding process is outlined in Fig. 3. This scheme is detailed step by step as follows:

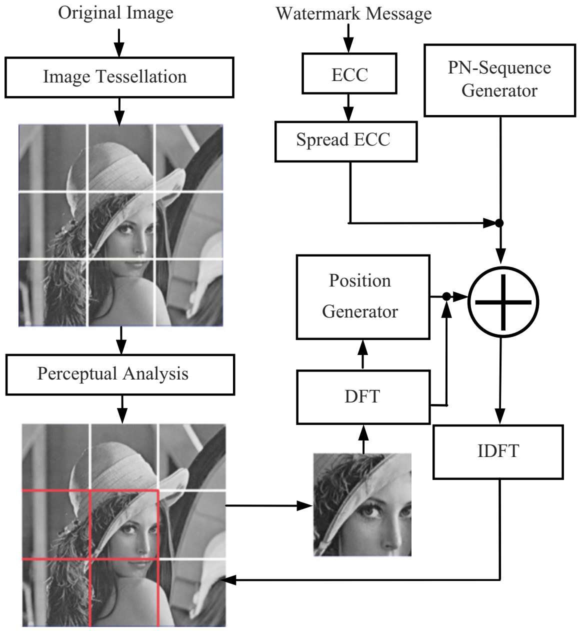

Fig. 3. Watermark embedding scheme.

1 *Image tessellation*. Evenly divide the 8-bit grayscale image of size 512 × 512 into 3 × 3 nonoverlapping subimages. The last several nondivisible rows and columns are not used for embedding.

2 *Perceptual analysis*. Apply our image-texture-based adaptive Harris corner detector to find the feature points in the original image by using a fixed 3 × 3 window. Choose the subimages with a large number of feature points (i.e., the number of feature points is more than a predetermined threshold Num) to be the embedding blocks. These blocks are perceptually high textured and are marked by red margins as shown in Fig. 3.

3 *Watermark generation and error correcting coding*. The watermark message $A_i$ is a bipolar sequence whose length is a factor of 4 (i.e., $|A_i| = 8$). It is then coded by error correcting coding (ECC) to generate an error correcting bipolar watermark bit sequence $\mathrm{ECA}_i$, whose length is a factor of 7 (i.e., $|\mathrm{ECA}_i| = 14$).

4 *Spread ECC*. Generate a spread error correcting bipolar watermark message bit sequence $W_j$ by repeating $\mathrm{ECA}_i$ $s$ times such that $W_j = \mathrm{ECA}_i$ for $is \leqslant j < (i+1)s$ where $s$ is the spreading factor (i.e., $s = \left\lfloor \frac{\mathrm{Len}}{|\mathrm{ECA}_i|} \right\rfloor$) and Len is the number of embedding positions. The $W_j$ will be further padded by zeros to ensure its length equals Len.

5 *PN-sequence generator*. Generate a PN-sequence $p_j \in \{1, -1\}$ with the same length as $W_j$ by using a secrete key.

6 For each perceptually high textured subimage SubA obtained from *step 2*:
   a. *DFT*. Apply global DFT to obtain FSubA.
   b. *Position generator*: Generate highly secure embedding positions in the mid-frequencies between $f_1$ and $f_2$ in the upper half plane of FSubA by using our one-way hash function.
   c. $\oplus$*operation*: Embed $W_j$ into each highly secure embedding position $(x_k, y_k)$ using:

$$F\text{SubA}(x_k, y_k)' = F\text{SubA}(x_k, y_k) + F\text{SubA}(x_k, y_k) \times G \times W_j \times p_j, \quad (8)$$

   where $G$ is the embedding strength. The same changes are carried out at center-based symmetric positions.
   d. *IDFT (Inverse DFT)*. Apply the IDFT to FSubA′ to obtain the watermark embedded subimage SubA′, which replaces the original subimage SubA.

Before the watermark is embedded, we apply the proposed image-texture-based adaptive Harris corner detector to find all important feature points in the original image. The results on three sample images are illustrated in Fig. 2. In our scheme, these important feature points represent the simplest pattern of the image features (i.e., image content), which are relatively robust against image distortions. Consequently, these important feature points are used to restore the probe image for reducing the synchronization errors and their positions will be saved for watermark detection. In addition, the threshold Num, the bipolar watermark message bit sequence $A_i$, the number of embedding positions Len, two secrete keys $n$ and $K$ for our one-way hash function in each subimage, two middle frequency ratios, and the secrete key for generating the PN-sequence will also be saved. Since the number of important feature points is regulated by the texture of the image, the storage is minimal compared to the cost of saving the original image itself. If all the information is compressed, the storage cost can be further minimized.

## 5. Watermark detection scheme

The block diagram of our watermark detection scheme is shown in Fig. 4. The watermark detection procedure does not need the original image. The important feature points are first extracted by applying the proposed image-texture-based adaptive Harris corner detector on the probe images. Two sets of triangles are then, respectively, generated by using the Delaunay tessellation on the saved important feature points and the important feature points found in the probe image. These two sets of triangles will be matched to determine the possible geometric transformations that the probe image has undergone. These determined geometric transformations will be further utilized to restore the probe image, so the synchronization errors between the extracted and original watermarks are minimized in the detection.

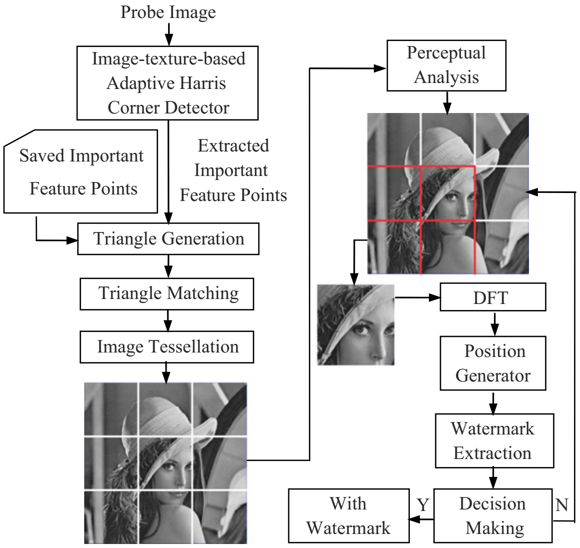

Fig. 4. Watermark detection scheme.

### 5.1. Image restoration: triangle generation

The triangle generation method plays an important role in watermark detection step. Though we can generate all possible triangles based on the important feature points to maximize the number of triangle matches between the two sets of triangles, doing so is very time-consuming. Suppose that we have 50 feature points on each image, we can generate 19 600 triangles in total on each set. If we exhaustively match all possible triangle pairs, there will be $19\,600 \times 19\,600 = 38\,41\,60\,000$ calculations. It probably will take a computer a couple of hours to finish such matching process. This is obviously unacceptable. Consequently, we need an effective and repeatable method to exclusively generate sufficient and useful triangles. Our choice is the

Delaunay tessellation due to its two attractive properties [38]:

- *Local property*. If a vertex disappears, the tessellation is only modified on the connected triangles.
- *Stability area*. Each vertex is associated with a stability area in which the tessellation pattern is not changed when the vertex is moved inside this area.

The Delaunay tessellation results of two sets of similar anchor points, with different displaced points, are shown in Fig. 5. As we can see, although losing or shifting an anchor point *a* will affect the triangle(s) connected to it, the tessellation pattern of other triangles will remain exactly the same. Thanks to the two properties of the Delaunay tessellation, we can always get an identical generation of triangles if the relative positions of anchors (i.e., important feature points) do not change. The Qhull algorithm [39] is utilized in our system to generate the important-feature-points-based triangles due to its fast speed and less memory constraints.

### 5.2. Image restoration: triangle matching

The triangle matching method follows the triangle generation. The matching criterion is based on the angle radians. That is, if two triangles have very similar angle radians (i.e., the angle difference is less than 0.01 rad), these two triangles are claimed to be likely matched. The possible geometric transformations are determined from the matched triangle pairs since the important-feature-points-based triangles in an image undergo the same transformation as the image itself. The detailed steps are:

1. Calculate the scaling factor SF by resizing the probe triangle to the same size as the target matched triangle.
2. Calculate the translation factor TF by registering one of the vertices of the matched triangle pair.
3. Calculate the rotation factor RF by aligning the other two unregistered vertices of the matched triangle pair.

These factors form a three-element-tuple (SF, TF, RF), where SF measures the scaling ratio up to a precision of 1/10, TF measures the translation in pixel numbers, and RF measures the rotation angle in an integer degree. Fig. 6 illustrates the above three steps by using a right triangle as an example. The three-tuple to represent this transformation is $(1.5, 15, 26^\circ)$. Since an image and the within triangles undergo exactly the same transformation, the majority of the identical three-element-tuple obtained from all the matched triangle pairs are used to restore the probe image to be aligned with the host image.

It is worth mentioning that sufficient important feature points can be found in case of non-geometric attacks as long as the probe image is not extremely low textured or undergoes JPEG compression with a quality factor of lower than 20%. The three-element-tuple (SF, TF, RF) for image restoration will be $(1, 0, 0)$. That is, the probe image has already been aligned with the host image. However, insufficient important feature points may be located when the probe image is extremely low textured or undergoes JPEG compression with a quality factor of lower than 20%. As a result, the restoration is likely to fail to find a majority of identical three-element-tuple for realignment. In this case, we will assume that there is no geometric attack and the three-element-tuple will be set to $(1, 0, 0)$ by default.

### 5.3. Watermark detection

The same embedding procedure is applied to the restored probe image to obtain the possible

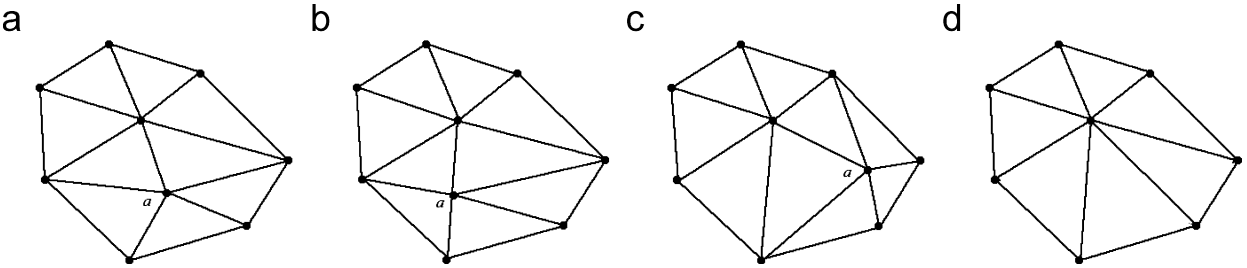

Fig. 5. Examples of the Delaunay tessellation of two similar point groups. (a) Original anchor points. (b) Point *a* is shifted in its stability area (i.e., a small shift around *a*). (c) Point *a* is shifted out of its stability area (i.e., a large shift far away from *a*) (d) Point *a* disappears.

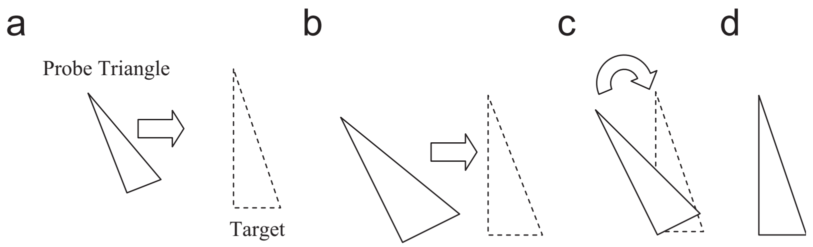

Fig. 6. The illustration diagram for finding the geometric transformations: (a) the matched triangle pair, (b) resizing result, (c) translation result and (d) rotation result.

watermark embedded DFT sequence FSubA$''_i$ for each potential embedding subimage $i$. The blind watermark retrieval scheme is then applied to extract the error correcting bipolar watermark message bit sequence ECA$'_i$. The parity bits are further used to correct 1 or 2 error bits, which may result from transmission noise. The error corrected possible watermark bit sequence ECA$''_i$ will be compared with the original error corrected watermark bit sequence ECA$_i$ to determine the presence of the watermark. That is, the number of matched bits in a potential embedding subimage is compared with a predefined threshold to determine whether the watermark is present in the probe image.

This predefined threshold is calculated based on the false-alarm probability that may occur in watermark detection. The Bernoulli trails are used to model ECA$''_i$ since every bit is an independent random variable. The probability of a $k$-bit match between $n$-bit extracted and original watermark bit sequences is calculated as:

$$p_k = \binom{n}{k} p^k (1-p)^{n-k}, \tag{9}$$

where $p$ is the success probability which indicates the possibility that the extracted bit matches the original watermark bit. We further simplify (9) by assuming $p$ to be 1/2:

$$p_k = \left(\frac{1}{2}\right)^n \left(\frac{n!}{k!(n-k)!}\right). \tag{10}$$

The false-alarm probability $P_{\text{false-alarm}}$ for each embedding subimage is computed as:

$$P_{\text{false-alarm}}(i) = \sum_{k_i=T_i}^{n} \left(\frac{1}{2}\right)^n \left(\frac{n!}{k_i!(n-k_i)!}\right). \tag{11}$$

That is, it is a cumulative probability of the cases that $k_i \geqslant T_i$, where $k_i$ and $T_i$ represent the number of matched bits and the threshold for each subimage $i$, respectively.

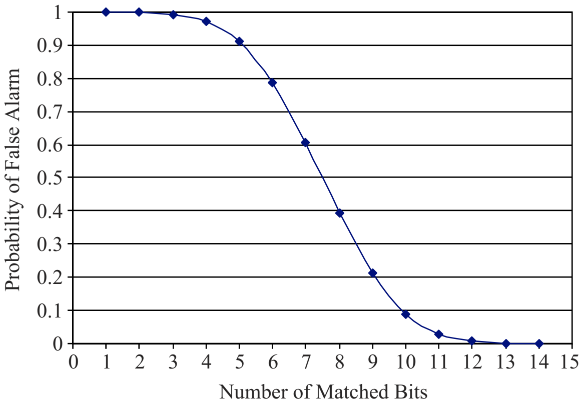

Fig. 7. The relationship between the false-alarm probability and the matched bits in an embedding subimage.

Fig. 7 shows the plot of $P_{\text{false-alarm}}$ against various threshold values. It demonstrates that the perfect match between the extracted and original watermark bit sequences in a single embedding subimage leads to a false alarm probability of $6.10 \times 10^{-5}$. That is, if the extracted and original watermark bit sequences are perfectly matched in a subimage, we claim the presence of the watermark with a false alarm probability of $6.10 \times 10^{-5}$. As a result, the watermark detection is based on $P_{\text{false-alarm}}$. In our system, we will check the perfect match in any embedding subimage to confidently claim the presence of the watermark since $6.10 \times 10^{-5}$ is a low false alarm probability.

## 6. Experimental results

To evaluate the performance of the proposed watermarking scheme, experiments have been conducted on various standard 8-bit grayscale images of size 512 × 512 and different kinds of attempting attacks. These standard images are either high textured, medium textured, or low textured as defined in (5).

### 6.1. Watermark invisibility

Watermark invisibility is evaluated on images of Lena, Baboon, and Pepper. These three images correspond to three texture categories. The PSNRs of these three watermarked images are 43.33, 44.06, and 37.62, respectively. In particular, the PSNR for each embedded subimage is: Lena (39.42, 37.02, and 36.6), Baboon (30.91, 32.75, 30.47, 31.87, 38.35, 32.78, 33.78, 34.60, and 32.64), and Pepper (42.65, 40.09, 39.13, and 40.52). These PSNR values are all greater than 30.00 db, which is the empirical value for the image without any perceivable degradation [40].

### 6.2. Important parameters

Several fixed parameters are used in our system. Specifically, the length of the bipolar watermark message bit sequence is 8. The length of the spread error correcting watermark sequence, which corresponds to the number of embedding positions in each perceptually high textured subimage, is 8192 for the image of size 512 × 512. This length linearly changes with the image size. However, if the length decreases to be less than 5000, we will use 2 × 2 nonoverlapping subimages as the possible embedding regions. The value of 5000 is an experimental lower bound to achieve enough spread spectrum positions for successful blind watermark retrieval. The embedding strength $G$ is 0.2, which ensures the appropriate embedded values for maintaining invisibility and facilitating detectable changes after image distortions on any subimage with the least textures. The embedding area in DFT domain is a ring with the inner and outer radii of 5% and 15% the size of the square subimage.

In addition to these fixed parameters, several image dependent parameters are also used in our system. Table 1 summarizes these adaptive parameters for the three different textured images, where *Ratio* is the factor for classifying image textures; *Type* is the texture decided by (5); $D$ is the diameter of the circular window used by our image-texture-based adaptive Harris corner detector; and SNum is the number of embedding subimages determined by perceptual analysis. In general, both $D$ and SNum are determined by the texture of the image. This is, the more complicate the texture, the larger the diameter $D$. Similarly, the more perceptually high texture each subimage has, the larger the value SNum.

Table 1
Several image texture dependant parameters

| | Lena | Baboon | Peppers |
|---|---|---|---|
| Ratio | 0.002 | 0.01 | 0.0013 |
| Type | Medium | High | Low |
| $D$ | 42 | 54 | 35 |
| Snum | 3 | 9 | 4 |

In summary, the same 8-bit bipolar watermark bit sequence is expanded to the same 14-bit error correcting bipolar watermark bit sequence. The same copy of this 14-bit watermark sequence is embedded in each perceptually high textured subimage at different embedding positions generated by our one-way hash function with different secret keys. The adaptive parameters are automatically determined based on the type of image textures. These image dependent parameters not only improve the accuracy in finding the image-content-based important feature points but also improve the robustness in resisting different geometric and common image processing attacks on different textured images.

### 6.3. Image restoration

The two steps of image restoration, the triangle generation and triangle matching, are crucial in the proposed watermarking scheme. In general, we apply the Delaunay tessellation on the important feature points to generate triangles and use angle degrees to find the matched triangles between the original and probe images. We further use these matched triangles to find the possible geometric attack in the three-element-tuple form since the transformation an image undergoes is directly reflected in the image-content-based triangles. Fig. 8 is an example of all possible matched triangles between the Lena image and the probe image, which has been distorted by a few geometric attacks as indicated in the captions of the figure. The matched triangles are indicated by the same colors. As shown in Fig. 8, we can always find enough number of matching triangles to restore the probe image to be aligned with the original image. This observation

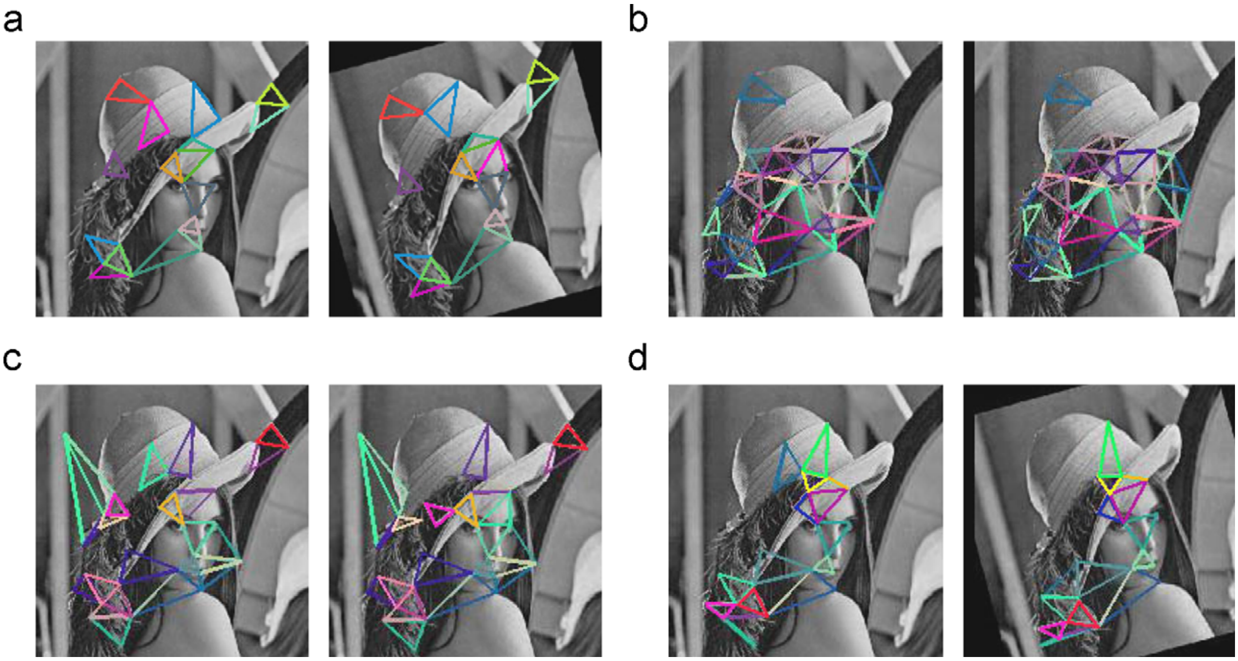

Fig. 8. The important-feature-points-based triangle matching under different distortions. (a) Matched triangles between the original image and the 15° rotated probe image. (b) Matched triangles between the original image and the 25 pixels vertical shifted probe image. (c) Matched triangles between the original image and the resized probe image with a factor of 0.8. (d) Matched triangles between the original image and the 15° rotated, 25 pixels vertical shifted, and 0.9 resized probe image.

Table 2
Ratios under different attacks

| Different images | Robust IFPs | Geometric attacks | | | |
|---|---|---|---|---|---|
| | | (a) | (b) | (c) | (d) |
| Lena | 53 | 14/16 | 35/36 | 10/11 | 14/16 |
| Baboon | 50 | 14/19 | 23/25 | 5/10 | 6/6 |
| Pepper | 59 | 26/27 | 21/22 | 8/10 | 6/9 |

further proves the effectiveness of the proposed approach.

Table 2 lists the number of important feature points on the three test images. It also lists the ratios between the number of matched triangle pairs for determining the geometric transformation and the total number of matched triangle pairs under the same four geometric attacks shown in Fig. 8. We observe that the number of important feature points is less than 60 for all the test images with different textures. This observation clearly demonstrates that our proposed adaptive Harris corner detector does regulate the number of important feature points. It also indicates that the cost of saving important feature points for watermark synchronization is minimal compared with the cost of saving the host image. Furthermore, all simulation results yield high ratios (most of them are higher than 70%), which indicate a high accuracy in finding the possible geometric transformation a probe image may undergo. When comparing the results between the images, it should be noted that the number of matched triangle pairs are not linearly related to the number of important feature points due to the sensitivity of the important feature points to different attacks (i.e., some important feature points may disappear, show up, or shift a bit in the attacked image). However, the two properties possessed by the Delaunay tessellation always ensure that there are enough matching triangles, which are shown as high ratios in Table 2, for restoring the probe image to be aligned with the original image.

### 6.4. Simulation results

Simulation of common image processing attacks and geometric distortions on a large variety of images has been performed to validate the proposed watermark scheme. Tables 3 and 4, respectively, demonstrate the simulation results of several common image processing and geometric attacks compared with Tang's method [17] on three images (i.e., 1 represents Lena, 2 represents Baboon, and 3 represents Pepper). All these attacks are intended to simulate a complete list of possible sample intentional and/or unintentional attacks occurred in the real world. Most image processing attacks and all the geometric attacks are performed by the benchmark software Stirmark 3.1. Since these two methods use different mechanisms to determine the presence of the watermark, the associated similarity scores are not highly significant to the comparison in this section. Hence, all the results will be recorded as "pass" or "fail" or "not available" to give a more intuitive comparison. For the ease of comparison, the results are listed side by side. That is, a method with more "passes" on its side has a better performance.

As shown in Table 3, our scheme performs better than Tang's method under common image processing attacks, such as median and Gaussian filtering, color quantization, and JPEG compression down to a quality factor of 30%. It also performs well under histogram equalization and some combined common image processing attacks including sharpening plus noise, image filtering plus JPEG compression, and image enhancement plus JPEG compression. This robustness against common image processing can be further improved with a stronger watermark embedding strength. However, a compromise between watermark robustness and invisibility needs to be considered for choosing the optimal embedding strength.

The watermark has also been correctly identified by our method under a variety of geometric attacks which Tang's method failed to handle as shown in blank cells and the cells containing "○" in Table 4. These geometric attacks include random relatively small and large rotations, scaling ratios, and any combination of RST attacks. However, our approach does not work well for the combined linear geometric transformation and the JPEG compression due to the possible loss of the important feature points.

Simulation results are also compared with the results yielded from the content-based scheme (CBS) [16] and the Digimarc watermarking tool available in Photoshop. In particular, Stirmark attacks such as small shearing, rotation 10°, scaling 0.8, and JPEG 50% have been performed to compare the performance. The scanning is not included due to lack of detailed information on this attack. The experimental results show that the proposed scheme can successfully detect the watermarks under these attacks. Therefore, our approach has comparable performance as the CBS and better performance than the Digimarc. In addition, our scheme can successfully resist several attacks that the CBS failed to handle. They are attacks of lower than 0.8 scaling on highly textured images and attacks of compression with a quality factor of lower than 50% as shown in Table 3.

Finally, we perform a variety of attacks on 105 8-bit watermarked grayscale images of size $512 \times 512$ using the benchmark software Stirmark 3.1. These images are evenly distributed with high, medium, and low textures according to (5). That is, the database contains 35 images for each texture level. The overall average PSNR value for these 105 watermarked images is 42.87 db. Table 5 summarizes the simulation results of 15 kinds of common image processing and geometric attacks on the 105 watermarked images. The simulated attacks correspond to a category of distortions including no attacks, translation, scaling, rotation, cropping up to 5%, linear geometric transform, row and column removal with a maximum of 20 rows and columns removed, median filtering, mean filtering, sharpening filtering, Gaussian filtering, histogram equalization, and JPEG compressions with quality factors of 40, 30, and 20. All the filtering operations use the maximum filter size of $3 \times 3$. Each distortion category contains two random attacks. Specifically, Table 5 lists the average detection rates of all images in each texture level under each distortion category using and without using ECC in order to compare their watermark detection performance. It clearly demonstrates that both our proposed scheme (i.e., the ECC-based approach) and its variant (i.e., the non-ECC-based approach) are robust against most geometric and image processing distortions and perform the worst for high textured images under the linear geometric and mean filtering attacks. Moreover, the average detection rates using ECC for all simulated geometric attacks are 92.34%, 87.25%, and 76.57% for medium, low, and high textured images, respectively. The average detection rates using ECC for all simulated image-processing attacks are 98.77%, 98.25%, and 87.62% for medium, low, and high textured images, respectively. The average detection rates for all

Table 3
The comparison between the proposed method and Tang's method under different common image processing

| Attack category | Attack name | Tang's result | | | Our result | | | Attack category | Attack name | Tang's result | | | Our result | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 1 | 2 | 3 | | | 1 | 2 | 3 | 1 | 2 | 3 |
| Watermark existence test | Watermarked image (no attack) | ● | ● | ● | ● | ● | ● | JPEG compression | 40% | ● | ● | | ● | ● | ● |
| Image filtering | Median filter 2 × 2 | | ● | | ● | ● | ● | | 30% | ● | ● | | ● | ● | ● |
| | Median filter 3 × 3 | | ● | | ● | ● | ● | Image filtering + JPEG 90% | Median filter 2 × 2 | ● | ● | | ● | ● | ● |
| | Sharpening 3 × 3 | ● | ● | ● | ● | ● | ● | | Median filter 3 × 3 | | | | ● | ● | ● |
| | Gaussian filtering 3 × 3 | ● | ● | | ● | ● | ● | | Sharpening 3 × 3 | ● | ● | ● | ● | ● | ● |
| | Mean filter 2 × 2 | ○ | ○ | ○ | ● | ● | ● | | Gaussian filtering 3 × 3 | ● | ● | | ● | ● | ● |
| | Mean filter 3 × 3 | ○ | ○ | ○ | ● | ● | ● | Image enhancement | Histogram equalization | ○ | ○ | ○ | ● | ● | ● |
| Quantization | Color quantization | ● | ● | | ● | ● | ● | Sharpening + Noise | Sharpening 3 × 3 + noise (scale = 0.1) | ○ | ○ | ○ | ● | ● | ● |
| Additive uniform noise | Scale = 0.1 | ● | ● | ● | ● | ● | ● | Image filtering + JPEG 50% | Median filter 2 × 2 | ○ | ○ | ○ | ● | ● | ● |
| | Scale = 0.15 | ● | ● | ● | ● | ● | ● | | Median filter 3 × 3 | ○ | ○ | ○ | ● | ● | ● |
| | Scale = 0.2 | | ● | | | ● | | | Mean filter 2 × 2 | ○ | ○ | ○ | ● | ● | ● |
| JPEG compression | 80% | ● | ● | ● | ● | ● | ● | | Mean filter 3 × 3 | ○ | ○ | ○ | ● | ● | ● |
| | 70% | ● | ● | ● | ● | ● | ● | | Sharpening 3 × 3 | ○ | ○ | ○ | ● | ● | ● |
| | 60% | ● | ● | | ● | ● | ● | | Gaussian filter 3 × 3 | ○ | ○ | ○ | ● | ● | ● |
| | 50% | ● | ● | ● | ● | ● | ● | Image enhancement + JPEG 50% | Histogram equalization | ○ | ○ | ○ | ● | ● | ● |

1 represents Lena, 2 represents Baboon, and 3 represents Pepper.
"●" indicates a "pass", blank cell means a "fail", and "○" indicates there is no result provided by the author.

Table 4
The comparison between the proposed method and Tang's method under different geometric distortions

| Attack category | Attack name | Tang's result | | | Our result | | | Attack category | Attack name | Tang's result | | | Our Result | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 1 | 2 | 3 | | | 1 | 2 | 3 | 1 | 2 | 3 |
| Row and column removal | 1 row and 5 columns | ● | ● | ● | ● | ● | ● | Rotation, cropping, and/or Scaling + JPEG 70% | 1° + Cropping + Scale | | ● | | ● | ● | ● |
| | 5 rows and 17 columns | | ● | | ● | ● | ● | | 1° + Cropping | ● | ● | | ● | ● | ● |
| Centered cropping | 5% | ● | ● | ● | ● | ● | ● | | 2° + Cropping | | | | ● | ● | ● |
| | 10% | ● | ● | ● | ● | ● | ● | | 5° + Cropping | | | | ● | ● | ● |
| Shearing | *x*–1%, *y*–1% | ● | ● | | ● | ● | ● | Linear geometric transform + JPEG 70% | (1.007, 0.01, 0.01, 1.012) | ● | ● | | ● | ● | ● |
| | *x*–0%, *y*–5% | ● | ● | | ● | ● | ● | | (1.010, 0.013, 0.009, 1.011) | ● | ● | ● | | ● | |
| | *x*–5%, *y*–5% | | ● | | | | | | (1.013, 0.008, 0.011, 1.008) | ● | ● | | | ● | |
| Rotation, cropping, and/or scaling | 1° + Cropping + Scale | | ● | ● | ● | ● | ● | Rotation | 15° | ○ | ○ | ○ | ● | ● | ● |
| | 1° + Cropping | ● | ● | ● | ● | ● | ● | | 35° | ○ | ○ | ○ | ● | ● | ● |
| | 2° + Cropping | | | | ● | ● | ● | | 210° | ○ | ○ | ○ | ● | ● | ● |
| | 5° + Cropping | | | | ● | ● | ● | Translation | [15,15] | ○ | ○ | ○ | ● | ● | ● |
| Linear geometric transform | (1.007, 0.01, 0.01, 1.012) | ● | ● | | ● | ● | ● | | [0, 25] | ○ | ○ | ○ | ● | ● | ● |
| | (1.010, 0.013, 0.009, 1.011) | ● | ● | | ● | ● | ● | | [25,0] | ○ | ○ | ○ | ● | ● | ● |
| | (1.013, 0.008, 0.011, 1.008) | ● | ● | | ● | ● | ● | Scaling | 90% | ○ | ○ | ○ | ● | ● | ● |
| Row and column removal + JPEG 70% | 1 row and 5 columns | ● | ● | ● | ● | ● | ● | | 80% | ○ | ○ | ○ | ● | ● | ● |
| | 5 rows and 17 columns | | ● | | ● | ● | ● | | 70% | ○ | ○ | ○ | ● | ● | ● |
| Centered cropping + JPEG 70% | 5% | ● | ● | ● | ● | ● | ● | Rotation, scaling, translation (RST) attacks | 5° + 80% + [0, 25] | ○ | ○ | ○ | ● | ● | ● |
| | 10% | ● | ● | ● | ● | ● | ● | | 15° + 90% + [2,25] | ○ | ○ | ○ | ● | ● | ● |
| Shearing + JPEG 70% | *x*–1%, *y*–1% | ● | ● | | ● | ● | ● | RST attacks + JPEG 70% | 5° + 90% + [5,5] | ○ | ○ | ○ | ● | ● | ● |
| | *x*–0%, *y*–5% | ● | ● | | ● | ● | ● | | 78° + 90% + [15,25] | ○ | ○ | ○ | ● | ● | ● |
| | *x*–5%, *y*–5% | | | | | | | | 10° + 80% + [10,10] | ○ | ○ | ○ | ● | ● | ● |

1 represents Lena, 2 represents Baboon, and 3 represents Pepper.
"●" indicates a "pass", blank cell means a "fail", and "○" indicates there is no result provided by the author.

Table 5
The average successful detection rates using ECC and without using ECC for three kinds of textured images under geometric and common image processing attacks using Stirmark

| Attack category | High textured images | | Medium textured images | | Low textured images | |
|---|---|---|---|---|---|---|
| | With ECC (%) | Without ECC (%) | With ECC (%) | Without ECC (%) | With ECC (%) | Without ECC (%) |
| No attacks | 100 | 100 | 100 | 100 | 100 | 100 |
| Translation | 100 | 91.43 | 100 | 97.14 | 92.85 | 91.43 |
| Scaling | 62.86 | 60.00 | 85.71 | 80.00 | 100 | 98.57 |
| Rotation | 65.71 | 62.86 | 97.14 | 91.43 | 88.57 | 85.71 |
| Cropping up to 5% | 100 | 97.14 | 94.29 | 91.43 | 100 | 97.14 |
| Linear geometric transform | 54.29 | 51.43 | 82.86 | 74.29 | 58.57 | 54.29 |
| Row and column removal | 74.29 | 71.43 | 94.29 | 94.29 | 84.28 | 81.43 |
| Median filtering | 62.86 | 60.00 | 97.14 | 94.29 | 100 | 95.71 |
| Mean filtering | 57.14 | 54.29 | 94.29 | 91.43 | 100 | 97.14 |
| Sharpening filtering | 100 | 100 | 100 | 97.14 | 100 | 100 |
| Gaussian filtering | 100 | 100 | 100 | 97.14 | 100 | 100 |
| Histogram equalization | 100 | 100 | 100 | 97.14 | 100 | 100 |
| JPEG 40% | 100 | 97.14 | 100 | 97.14 | 100 | 100 |
| JPEG 30% | 100 | 97.14 | 100 | 95.71 | 100 | 100 |
| JPEG 20% | 94.29 | 91.43 | 97.14 | 94.29 | 100 | 98.57 |

simulated attacks are 96.28%, 94.90%, and 84.62% for medium, low, and high textured images, respectively. The overall average detection rate for all images under all simulated attacks is 91.93%. Compared with the simulation results obtained without using ECC, our proposed scheme improves the average detection rates for all simulated geometric attacks by 5.92%, 3%, and 5.51% for medium, low, and high textured images and improves the average detection rates for all simulated image processing attacks by 2.83%, 1.31%, and 2.22% for medium, low, and high textured images, respectively. The average detection improvement for all simulated attacks is 3.59%, 1.73%, and 3.01% for medium, low, and high textured images, respectively. The overall average detection improvement for all images under all simulated attacks is 2.77%. The detailed comparison illustrates that the ECC-based approach improves the average detection rates for watermarked images under both geometric and image processing attacks. However, it achieves more improvement for the geometric attacks since imperfect image restoration yields relatively more mismatches between the extracted and embedded watermark sequences and the ECC may correct these small errors.

In summary, our proposed watermarking scheme outperforms the peer content-based schemes [16,17]. It yields positive detection results for most images with low, medium, and high textures under different geometric distortions and common image processing attacks. This robustness is mainly due to the following three factors:

1. The image-texture-based adaptive Harris corner detector is capable of finding the important feature points for different textured images to roughly represent invariant references to various geometric transformations.
2. The Delaunay-tessellation-based triangle generation and matching can detect and correct the possible geometric attacks on the watermarked image even when some important feature points disappear, show up, or shift a bit in the attacked image. In particular, the triangle generation is able to eliminate the effect of the fake important feature points showed up in high textured images and minimize the synchronization errors between the extracted and original watermarks. The

triangle matching process requires substantially fewer interpolation operations than other peer systems [16,17] and therefore significantly improves the accuracy in extracting and detecting watermarks.

3. The DFT domain itself is robust to translation and moderate cropping so it can accommodate the cropping attacks and further compensate the slightly inaccurate important feature points based geometric correction.

The success of the proposed watermarking scheme is also due to the following three features: (1) The perceptually high textured subimage embedding scheme helps our watermarking system to survive some localized image attacks in Stirmark. (2) The image-texture-based adaptive Harris corner detector is capable of regulating the important feature points for different textured images and therefore enables our scheme to perform reasonably well for high textured images, which cannot be handled by other peer systems [16,17]. (3) The robustness of the spread spectrum embedding and detection makes our scheme more resistant to common image processing attacks.

However, our scheme does not perform well for the extremely low textured images due to the insufficient important feature points. It also fails the JPEG compression with a quality factor of lower than 20% due to the missing important feature points resulted from high compression. Furthermore, it fails under mirror and flip attacks since the algorithm cannot correctly match two triangles which are mirrored or flipped.

## 7. Conclusions

In this paper, we propose a novel and effective content-based robust watermarking approach. The major contributions consist of:

1. Image-texture-based adaptive image content extraction: This extraction method is capable of finding the important feature points, which are robust against geometric attacks, for images with high, medium, and low textures.
2. Error correcting bipolar watermark bit sequence: This sequence can correct one or two transmission errors in the extracted watermark sequence to improve the detection accuracy with a low false alarm and false negative probability.
3. Image dependent perceptually high textured subimage selection for embedding: These embedding subimages carry the same copy of the error correcting bipolar watermark bit sequence to improve the robustness of transmitted watermark bits. They also aid our watermarking scheme in surviving some localized image attacks in Stirmark.
4. Spread-spectrum-based blind watermark embedding and retrieval in DFT domain: The spread spectrum scheme makes our system more resistant to common image processing attacks. The DFT domain ensures more resistant to translation and moderate cropping. The blind embedding and retrieval scheme does not require any information about the original image at the detection stage.
5. Delaunay-tessellation-based triangle generation and matching: This scheme can efficiently eliminate the effect of inaccurate important feature points and correctly determine the possible transformation a probe image may undergo with fewer interpolation operations.

The proposed method is robust against a wide variety of tests as indicated in the experimental results. In particular, it is more robust against JPEG compression and the combination of the geometric distortions with large scaling ratios and rotations than other content-based watermarking techniques. It works successfully for images with high, medium, and low textures. Our approach can be further improved by developing a more reliable feature extraction method under severe geometric distortions and a more efficient and accurate triangle generation and matching method. In the real world, this watermarking technique can be applied to a lot of different areas, such as photograph, audio, and video. One example is that computer graph artists can use this method to impose an invisible watermark into their works for identifying the ownership.

## References

[1] F. Peticolas, R. Anderson, M. Kuhn, Attacks on copyright marking systems, in: Proceedings of the Second Workshop Information Hiding, Portland, OR, April 1998, pp. 218–238.

[2] S. Pereira, J.J.K. O'Ruanaidh, F. Deguillaume, G. Csurka, T. Pun, Template based recovery of Fourier-based watermarks using log-polar and log-log maps, in: Proceedings of the IEEE International Conference Multimedia Computing Systems, vol. 1, Florence, Italy, June 1999, pp. 870–874.

[3] S. Pereira, T. Pun, Robust template matching for affine resistant image watermarks, IEEE Trans. Image Process. 9 (6) (2000) 1123–1129.

[4] Digimarc Corporation, US patent 5,822,436, Photographic Products and Methods Employing Embedded Information.

[5] J.J.K. O'Ruanaidh, T. Pun, Rotation, scale, and translation invariant digital image watermarking, in: Proceedings IEEE International Conference Image Processing, Santa Barbara, CA, 1997, pp. 536–539.

[6] J.J.K. O'Ruanaidh, T. Pun, Rotation, scale, and translation invariant spread spectrum digital image watermarking, Signal Process. 66 (3) (1998) 303–317.

[7] D. Zheng, J. Zhao, A. El Saddik, RST-invariant digital image watermarking based on log-polar mapping and phase correlation, IEEE Trans. Circuits Syst. Video Technol. 13 (8) (2003) 753–765.

[8] C.Y. Lin, M. Wu, J.A. Bloom, I.J. Cox, M.L. Miller, Y.M. Lui, Rotation, scale, and translation resilient watermarking for images, IEEE Trans. Image Process. 10 (5) (2001) 767–782.

[9] M. Alghoniemy, A.H. Tewfik, Geometric distortion correction through image normalization, in: Proceedings of IEEE International Conference Multimedia Expo, vol. 3, 2000, pp. 1291–1294.

[10] M. Alghoniemy, A.H. Tewfik, Image watermarking by moment invariants, in: Proceedings of IEEE International Conference Image Processing, vol. 2, Janual 2000, pp. 73–76.

[11] M. Alghoniemy, A.H. Tewfik, Geometric invariance in image watermarking, IEEE Trans. Image Process. 13 (2) (2004) 145–153.

[12] H.S. Kim, H.K. Lee, Invariant image watermark using Zernike moments, IEEE Trans. Circuits Syst. Video Technol. 13 (8) (2003) 766–775.

[13] Y. Xin, S. Liao, M. Pawlak, Geometrically robust image watermarking via pseudo-Zernike moments, in: Proceedings of the Canadian Conference Electrical and Computer Engineering, vol. 2, May 2004, pp. 939–942.

[14] A. Herrigel, S. Voloshynovskiy, Y.B. Rytsar, Watermark template attack, in: Proceedings of SPIE Security and Watermarking of Multimedia Contents III, vol. 4314, January 2001, pp. 394–405.

[15] M. Kutter, S.K. Bhattacharjee, T. Ebrahimi, Toward second generation watermarking schemes, in: Proceedings of IEEE International Conference Image Processing, vol. 1, Kobe, Japan, October 1999, pp. 320–323.

[16] P. Bas, J.M. Chassery, B. Macq, Geometrically invariant watermarking using feature points, IEEE Trans. Image Process. 11 (9) (2002) 1014–1028.

[17] C.W. Tang, H.M. Hang, A feature-based robust digital image watermarking scheme, IEEE Trans. Signal. Process. 51 (4) (2003) 950–959.

[18] S. Bhattacharjee, M. Kutter, Compression tolerant image authentication, in: Proceedings of the IEEE International Conference Image Processing, vol. 1, 1998, pp. 435–439.

[19] D. Delannay, B.M. Macq, Method for hiding synchronization marks in scale- and rotation-resilient watermarking schemes, in: Proceedings of SPIE Security and Watermarking of Multimedia Contents IV, vol. 4675, 2002, pp. 548–554.

[20] K. Su, D. Kundur, D. Hatzinakos, Spatially localized image-dependent watermarking for statistical invisibility and collusion resistance, IEEE Trans. Multimedia 7 (1) (2005) 52–66.

[21] X. Kang, J. Huang, Y.Q. Shi, Y. Lin, A DWT-DFT composite watermarking scheme robust to both affine transform and JPEG compression, IEEE Trans. Circuits Syst. Video Technol. 13 (8) (2003) 776–786.

[22] D. Simitopoulos, D.E. Koutsonanos, M.G. Strintzi, Robust image watermarking based on generalized Radon transformations, IEEE Trans. Circuits Syst. Video Technol. 13 (8) (2003) 732–745.

[23] M.Y. Wu, Y.K. Ho, A robust object-based watermarking scheme based on shape self-similarity segmentation, in: Proceedings of Fifth International Symposium Multimedia Software Engineering, December 2003, pp. 110–113.

[24] C. Harris, M. Stephen, A combined corner and edge detector, in: Proceedings of Fourth Alvey Vision Conference, Manchester, 1988, pp. 147–151.

[25] J. Devars, C. Achard-Rouquet, E. Bigorgne, Un detecteur de point carateristiques sur des images multispectrales extension vers un detecteur sub-pixellique, in: Proceedings of GRETSI'99, September 1999, pp. 627–630.

[26] S.M. Smith, J.M. Brady, SUSAN—a new approach to low level image processing, Int. J. Comput. Vis. 23 (1) (1997) 45–78.

[27] C. Schmid, R. Mohr, C. Bauckhage, Comparing and evaluating interest points, in: Proceedings of the Sixth International Conference on Computer Vision, Bombay, 1998, pp. 230–235.

[28] F. Heitger, L. Rosenthaler, R. von der Heydt, E. Peterhans, O. Kuebler, Simulation of neural contour mechanism: from simple to end-stopped cells, Vis. Res. 32 (5) (1992) 963–981.

[29] W. Forstner, A framework for low level feature extraction, in: Proceedings of the Third European Conference on Computer Vision, Stockholm, Sweden, May 1994, pp. 384–394.

[30] R. Horaud, T. Skordas, F. Veillon, Finding geometric and relational structures in an image, in: Proceedings of the First European Conference on Computer Vision, Antibes, France, 1990, pp. 374–384.

[31] J. C. Cottier, Extraction et appariements robustes des points d'int6rkt de deux images non etalonnees, Technical Report, LIFA-IMAG-INRIA, Rhone-Alpes, 1994.

[32] H. Moravec, Obstacle avoidance and navigation in the real world by a seen robot rover, Robotics Institute, Carnegie-Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-RI-TR-3, September 1980.

[33] F.J. MacWilliams, N.J. Sloane, Theory of Error Correcting Codes, North-Holland, Amsterdam, Netherlands, 1977.

[34] I.J. Cox, J. Kilian, F.T. Leighton, T. Shamoon, Secure spread spectrum watermarking for multimedia, IEEE Trans. Image Process. 6 (12) (1997) 1673–1687.

[35] M.S. Hwang, C.C. Chang, K.F. Hwang, A watermarking technique based on one-way hashing functions, IEEE Trans. Consumer Electron. 25 (2) (1999) 286–294.

[36] F. Hartung, B. Girod, Watermarking of uncompressed and compressed video, Signal Process. 66 (3) (1998) 283–301.

[37] S. Pranata, Y.L. Guan, H.C. Chua, BER formulation for the blind retrieval of MPEG video watermark, Lecture Notes in Computer Science, vol. 2613, pp. 91–104, 2003.

[38] E. Bertin, S. Marchand-Maillet, J.M. Chassery, Optimization in Voronoi Diagrams, Kluwer, Norwel, MA, 1994, pp. 209–216.

[39] C.B. Barber, D.P. Dobkin, H.T. Huhdanpaa, The Quickhull algorithm for convex hulls, ACM Trans. Math. Software 22 (4) (1996) 469–483.

[40] M.S. Hsieh, D.C. Tseng, Perceptual digital watermarking for image authentication in electronic commerce, Electron. Commerce Res. 4 (2004) 157–170.