A Robust DCT-Based Digital Watermarking Scheme Using Important Feature Points

Xiaojun Qi and Ji Qi

Department of Computer Science, Utah State University
Logan, UT, 84322-4205, USA.
Xiaojun.Qi@usu.edu and *jiqi@cc.usu.edu*

Abstract: This paper proposes a human visual system based Direct Current (DC) components embedding method to adaptively embed secret information bits into the blockwise Discrete Cosine Transform (DCT) domain. This embedding scheme achieves good robustness against common image processing attacks and provides a high perceptual capacity for watermark embedding. A spread spectrum technique codes the watermark to ensure a large measure of security against unintentional and intentional attacks. The proposed scheme further achieves the Rotation, Scaling, and Translation (RST) resilience by restoring the synchronization between host and probe images. Specifically, an image-texture-based adaptive Harris corner detector locates the Important Feature Points (IFPs) with high geometric significance. Those IFPs are used as anchor points for the Delaunay triangle tessellation. We combine a triangle-based image restoration method and an image normalization technique to restore the synchronization between host and probe images. The watermark is detected based on the correlation between the recovered and embedded watermarks. Our extensive experiments demonstrate that the proposed watermarking approach is robust against various RST attacks and some common image processing operations such as JPEG compression, median filtering, and Gaussian filtering. It also achieves a better performance as compared with five peer systems in the literature.

Keywords: Geometrically invariant digital watermarking, important feature points, image-texture-based adaptive Harris corner detector, Delaunay-tessellation-based triangle generation, direct-current-components-based embedding

1. Introduction

The rapid development of information technology, especially information digitization, enables people to access huge amounts of information in recent years. The digitized information can further be reproduced and instantaneously distributed at basically no cost with the aid of the high speed Internet and the immediate availability of computing resources. However, large-scale unauthorized copying may also emerge and therefore undermine the profits of the photography, music, file, and book publishing industries. As a result, patent protection of the intellectual properties attracts more and more attention from researchers and patent protection organizations. A possible technical solution to this is digital watermarking, a label or mark embedded into digital media to automatically identify its copyright owner or its buyer, and further detect and possibly prosecute copyright infringement.

In general, any watermarking technique requires several properties, including transparency, robustness, trustworthy detection, and computational efficiency [1]. Among those properties, transparency and robustness are the most important. Although the research in transparency [2-5] has achieved promising results, the robustness, especially against geometric distortions, still remains as a hard problem to tackle since synchronization errors can be induced in watermark detection and decoding. Several state-of-the-art watermarking schemes have been developed to counterattack geometric distortions. These schemes can be roughly divided into invariance-domain-based, template-based, moment-based, and feature-based algorithms.

Invariance-domain-based watermarking algorithms [6-8] generally provide a Rotation, Scaling, and Translation (RST) invariant domain for embedding watermarks and maintaining synchronization under affine transforms. However, interpolation errors in both forward and inverse transformation may lead to the inaccuracy problem, and the resampling and integration may cause aliasing.

Template-based watermarking algorithms [9, 10] embed structured templates in Discrete Fourier Transform (DFT) domain to identify the geometric transformation and therefore assist watermark synchronization in the detection process. However, the image-independent characteristic features of the

templates can be exploited to derive a new attack [11] to destroy the structured templates without any prior knowledge of their synchronization pattern. Consequently, even if the watermark remains, it cannot be extracted.

Moment-based watermarking algorithms [12-15] utilize geometric moments or normalized RST invariant Zernike moments to provide a solution to the geometric invariance problem. However, they are normally sensitive to cropping and aspect ratio changes. Furthermore, perfect invariance cannot be achieved due to the discretization errors.

Feature-based watermarking algorithms [16-22] use image dependent feature points as a content descriptor to represent invariant reference points for both watermark embedding and detection. Bas *et al.* [16] use the Harris detector for feature extraction. The feature points are mixed with a Delaunay tessellation to mark each watermark embedding triangle. The original watermark triangles will be warped during the detection to correlate with the marked triangles. Similarly, Seo and Yoo [17, 18] extract feature points using the Harris-Laplace detector and decompose the image into disjointed local circular or elliptical regions for watermark embedding and extraction. In Lee's approach [19], the Scale-Invariant Feature Transform (SIFT) is used to determine the local circular regions for watermarking. The simulation results from all of the above methods show that the robustness of the scheme depends on the capacity for preserving feature points after geometric transformation, especially on images with more texture and images with less texture and large homogeneous areas. Furthermore, all these methods embed the watermark in spatial domain after geometric normalization according to the shapes of the regions. Consequently, watermark robustness is not satisfactory and the feature point based transform domain watermarking schemes have been proposed. Tang and Hang [20] apply the Mexican hat wavelet scale interaction method to extract feature points. The watermark is embedded and extracted in the DFT domain of the normalized disks centered at the extracted feature points. Wang *et al.* [21] improved Tang's method by using the scale invariant Harris-Laplace detector to find the radius of each circular region. However, Tang's scheme performs well under only mild geometric distortions and certain common image processing attacks. The watermark capacity of both schemes is only 16 bits, which may restrict their practical applications.

In this paper, we propose a robust invisible digital watermarking scheme for images. This scheme combines the advantages of our novel important feature extraction, Human Visual System (HVS), image normalization, Direct Current (DC) components-based blind watermark embedding and retrieval, Delaunay-tessellation-based triangle generation, and triangle-based image restoration to resist image geometric distortions and common image processing attacks. Section 2 describes the watermark embedding procedure of the proposed method. Section 3 covers the details of our watermark extraction procedure. Section 4 shows the simulation results by comparing our scheme with five approaches in terms of robustness against both geometric distortions and common image processing attacks. In addition, we also demonstrate the performance of our proposed scheme on 105 images of various textures under different Stirmark attacks. Section 5 concludes this presentation and discusses the direction for future work.

2. Watermark Embedding Procedure

The block diagram of our watermark embedding scheme is shown in Figure 1. This embedding scheme is optimized in the sense that it can embed a robust watermark into the host image without any noticeable distortion. The watermarked image is also robust against geometric distortions and common image processing attacks. Each embedding step is detailed in the following subsections.

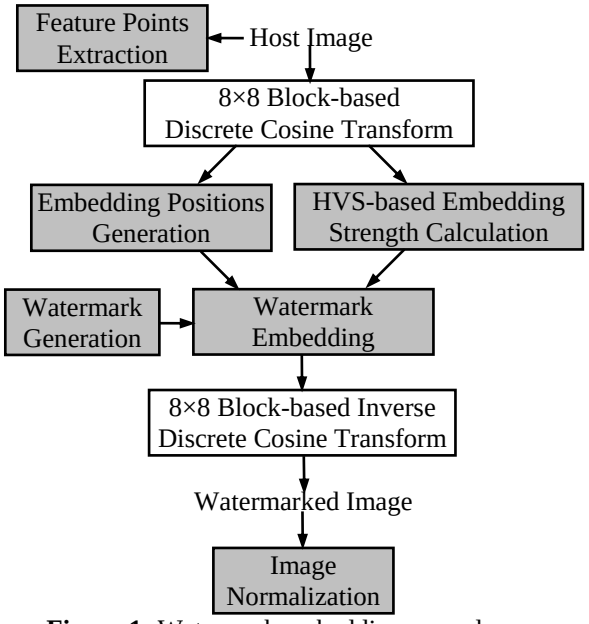


Figure 1: Watermark embedding procedure.

2.1. Feature Points Extraction

Extracting 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 feature points that are perceptually significant and can thus potentially resist various common image processing and geometric distortions. These image-content-bounded feature points can be further used as anchor points in watermark detection.

In our proposed system, we develop an image-texture-based adaptive Harris corner detector to locate the feature points. The choice of the Harris corner detector [23] is mainly based on the following two reasons: 1) It is the most robust compared with other commonly used detectors [16]. 2) It is the most stable compared with the other detectors with regards to the repeatability of the detector when the detected corner points are used for matching purposes [24]. The common Harris corner detector [23] first defines a shape-factor-based matrix at each position (i, j):

$$M(i,j) = \begin{bmatrix} A_{i,j} & C_{i,j} \\ C_{i,j} & B_{i,j} \end{bmatrix} = \begin{bmatrix} \sum_{m,n} G_x(m,n)^2 & \sum_{m,n} G_x(m,n)G_y(m,n) \\ \sum_{m,n} G_x(m,n)G_y(m,n) & \sum_{m,n} G_y(m,n)^2 \end{bmatrix} \quad (1)$$

where $G_x(m, n)$ and $G_y(m, n)$ are the horizontal and vertical gradients at the neighborhood areas (m, n)'s centered at each position (i, j) in an image, respectively. The corner response value $R(i, j)$ at each position (i, j) is further computed by:

$$R(i,j) = det\big(M(i,j)\big) - k\big[trace\big(M(i,j)\big)\big]^2 = \big(A_{i,j}B_{i,j} - C_{i,j}^2\big) - k\big(A_{i,j}+B_{i,j}\big)^2 \quad (2)$$

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

In our proposed feature points extraction scheme, we improve the performance of the common Harris corner detector by applying some pre-processing techniques to reduce the noise effect and regulating the density of the feature points based on the image texture. This extraction procedure is as follows:

1. Apply a 3×3 Gaussian low-pass filter to the original image to avoid corners due to noise.
2. Apply a rotationally symmetric 3×3 Gaussian low-pass filter with the standard deviation of 0.5 to three gradient images ($A_{i,j}$, $B_{i,j}$, and $C_{i,j}$) to achieve additional resistance to noise.
3. Calculate $R(i, j)$ within a circular window, which is at the image center and covers the largest area of the original image. The resulting $R(i, j)$'s reduce the effect of image-center-based rotation attacks.
4. Search for Important Feature Points (IFPs) based on the local maxima:
   $$\{R(i,j) | R(i,j) > T \cap R(i,j) \geq R(u,v), \forall (u,v) \in V_{i,j}\} \quad (3)$$
   where T is a predefined threshold value to extract a desired number of corner points, and $V_{i,j}$ represents a circular neighborhood centered at (i, j).

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 since a small window makes the feature points concentrate on textured areas and a large window tends to isolate the feature points. That is, the smaller window yields more feature points and the larger window yields fewer feature points. It therefore follows that one can increase the good matches of feature points on both original and probe images by decreasing the window size. However, this is done at the price of a proportional increase in the total number of corner points to be analyzed. 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:

$$D = \sqrt{\frac{wh}{np}} \quad (4)$$

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 image texture. It is set to be 1.5, 2.5, and 3.5 for images with high, medium, and low textures, respectively. The texture of the image is roughly classified by the following criterion:

$$Texture = \begin{cases} High, & ratio \geq 0.01 \\ Medium, & ratio \geq 0.002 \\ Low, & ratio \geq 0.0001 \end{cases} \quad (5)$$

where *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 fixed 3×3 neighborhood window.

With an adaptive and optimized window size for the Harris corner detector, we can make sure a certain amount of corner points always are detected, neither too many, nor too few. By doing this, the computational cost of the feature points extraction and the triangle matching can be balanced in watermark detection. Figure 2 demonstrates the extracted IFPs by applying our image-texture-based adaptive Harris corner detector on 3 images with different textures. The resultant feature points are shown as large white squares for display purpose. It clearly shows that the number of IFPs is regulated by the image texture. That is, medium and low textured images such as Lena and Pepper have enough IFPs for watermark detection process to reduce the synchronization errors. Similarly, high textured images such as Baboon do not have overwhelming number of IFPs as suffered by other feature-based methods. On the contrary, they possess sufficient IFPs for watermark detection process to achieve self-synchronization. It is worthwhile to mention that extremely low textured images (i.e., *ratio*<0.0001) will not be considered for copyright protection due to insufficient IFPs for self-synchronization.

2.2. Embedding Positions Generation

Before embedding the watermark bit sequence, we divide host image I into non-overlapping 8×8 blocks. Each block is separately transformed by the Discrete Cosine Transform (DCT) and the final transformed

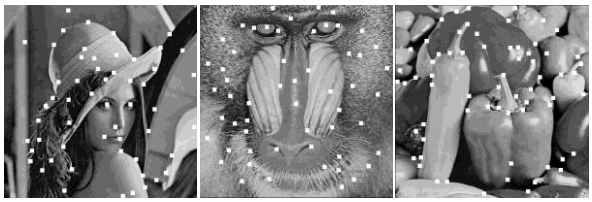


Figure 2: Important feature points extracted by our image-texture-based adaptive Harris corner detector.

image is named as I_{dct}. Figure 3 shows the proposed strategy for generating embedding positions. That is, every 4 spatially adjacent 8×8 DCT blocks is grouped together and embedded with a single watermark bit. Each of these 4 block groups is called one embedding unit, which is shown as either gray or white. One example of the embedding units is shown in Figure 3 as the group of blocks A, B, C, and D. The embedding positions in each embedding unit are the DC component (i.e., the top left value of each block shown as a check mark) of each 8×8 DCT block. Specifically, a single watermark bit is repetitively embedded into each of the four embedding positions to increase the redundancy of the embedded information. Based on the above strategy, the maximum number of embedding units corresponds to the maximum length of the watermark bit sequence and is 1024 for an image of size 512×512.

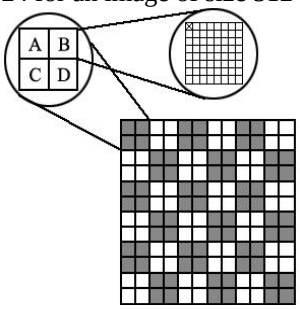


Figure 3: Example of embedding units.

Even though low-frequency Amplitude Current (AC) components are widely considered to be good watermark embedding sites and DC components are explicitly excluded from conventional watermark embedding, we argue that DC components can be more suitable for watermarking than any AC components based on the following four reasons: 1) DC components provide a higher perceptual capacity [3] for watermark embedding since the magnitude of the DC component is generally much larger than that of any AC component. 2) According to Weber's law [25], DC components can avoid block artifacts under the invisibility constraint since they can be modified by a larger quantity than any AC component. 3) According to the theory of signal processing, DC components tend to be changed less than AC components under common image processing attacks that a watermarked image may encounter. 4) The DC-components-based embedding method tends to yield better robustness and visual quality than common AC-components-based embedding methods by carefully selecting the embedding strength [26]. Such an advantage is mainly due to the fact that the capacity of a DC component normally is 10% of its intensity and the absolute value of a DC component is higher than the total intensity values of all the AC components.

2.3. HVS-based Embedding Strength Calculation

The Human Visual System (HVS) model plays a critical role in our embedding scheme since DC components are chosen as the embedding positions. Without an appropriate HVS model, the embedding may have severe blocking effects on the watermarked image. Consequently, we adopt Watson's DCT-based visual model [27] to estimate the magnitude that the DC component of each 8×8 block may be changed without introducing a Just Noticeable Difference (JND). The chosen Watson's model estimates the sensitivity of human eyes to the changes of the brightness in each 8×8 DCT block since brighter regions are able to absorb larger changes without becoming noticeable. It also gives a quantitative measure of the embedding capacity of each DCT block by calculating the luminance and contrast masks. The detailed procedure for computing the HVS-based embedding strength, which determines the magnitude to be changed on the DC component of each 8×8 DCT block, is as follows:

1. Transform the original image into an 8×8 block-based DCT image I_{dct}.
2. Compute the luminance masked threshold $t_L[i,j,k]$ for each 8×8 block k by:
$$t_L[i,j,k] = t[i,j]\left(\frac{C_0[0,0,k]}{C_{0,0}}\right)^r \quad (6)$$
where $t[i, j]$ is the DCT frequency sensitivity at position (i, j) as shown in Table 1 [28], $C_0[0, 0, k]$ is the DC coefficient of the k^{th} block, $C_{0,0}$ is the average of the DC coefficients in all blocks, r is a constant with an empirical value of 0.649, $0 \le i,j \le 7, 1 \le k \le N$, and N is the total number of blocks in an image.
3. Compute the contrast masked threshold $s[i, j, k]$ for each 8×8 block k by:
$$S[i,j,k] = max\{t_L[i,j,k]|C_0[i,j,k]^{0.7}|t_L[i,j,k]^{0.3}\} \quad (7)$$
where $t_L[i, j, k]$ is the luminance masked threshold for each term of DCT block k, and $C_0[i, j, k]$ is the DCT coefficient at position (i, j) in block k.
4. Compute the capacity of each block, S_k, by:
$$S_k = \sum_{i=0}^{7}\sum_{j=0}^{7} s[i,j,k] \quad (8)$$
where $s[i, j, k]$ is the contrast masked threshold at position (i, j) of block k.
5. Decide the appropriate embedding strength α for each block k by:
$$\alpha = \begin{cases} 100, & S_k > 0.5(S_{mean} + S_{max}) \\ 50, & Otherwise \end{cases} \quad (9)$$
where S_{mean} and S_{max} are the mean and maximum capacities among all the blocks in an image, respectively. Here, we experimentally

choose α to be 50 or 100. That is, the smaller embedding strength (i.e., 50) can achieve good invisibility in smooth embedding blocks (i.e., areas with low complexity) and the larger embedding strength (i.e., 100) can achieve good invisibility and better robustness in rich high-frequency embedding blocks (i.e., areas with high complexity).

Table 1: DCT frequency sensitivity table

Row \ Col	0	1	2	3	4	5	6	7
0	1.40	1.01	1.16	1.66	2.40	3.43	4.79	6.56
1	1.01	1.45	1.32	1.52	2.00	2.71	3.67	4.93
2	1.16	1.32	2.24	2.59	2.98	3.64	4.60	5.88
3	1.65	1.52	2.59	3.77	4.55	5.30	6.28	7.60
4	2.40	2.00	2.98	4.55	6.15	7.46	8.71	10.17
5	3.43	2.71	3.64	5.30	7.46	9.62	11.58	13.51
6	4.79	3.67	4.60	6.28	8.71	11.58	14.50	17.29
7	6.56	4.93	5.88	7.60	10.71	13.51	17.29	21.15

This 5-step procedure ensures the noticeable distortion will be avoided by using the adaptive embedding strengths. Specifically, the higher value of $C_0[0, 0, k]$ in block k leads to the larger luminance masked threshold. This matches with the observation that the brighter blocks of an image are able to absorb larger changes without becoming noticeable. Furthermore, the contrast masked threshold estimates the amounts that each individual term in an 8×8 DCT block may be changed before exceeding the JND since $s[i, j, k]$ depends on either the energy or the luminance masked threshold at the $(i, j)^{th}$ frequency. The final step 5 guarantees that a larger embedding strength is used in high capacity regions than in low capacity regions, where the region capacity is measured by the summation of all the contrast masked thresholds in the corresponding block.

2.4. Watermark Generation and Embedding

In our proposed system, the watermark message is a pseudo-random sequence M generated by a secret key k, which is kept by the owner to ensure the security of the message. The advantages of using a pseudo-random sequence are [3]: 1) It is more secure and arouses less suspicion in attackers. 2) It is robust against noise. 3) It achieves error free transmission near or at the limits set by Shannon's noisy channel coding theorem. A 1024-bit watermark message M is generated for an 8-bit 512×512 grayscale image. This watermark is then embedded into the DC components of the embedding units in the host image. If the k^{th} bit of M is 1, it is embedded as:

$$DC'_{k,i} = \begin{cases} DC_{k,i} - (DC_{k,i} \bmod \alpha) + .75\alpha & if\ (DC_{k,i} \bmod \alpha) \geq .25\alpha \\ DC_{k,i} - .25\alpha - [(DC_{k,i} - .25\alpha) \bmod \alpha] + .75\alpha & Otherwise \end{cases} \quad (10)$$

If the k^{th} bit of M is -1, it is embedded as:

$$DC'_{k,i} = \begin{cases} DC_{k,i} - (DC_{k,i} \bmod \alpha) + .25\alpha & if\ (DC_{k,i} \bmod \alpha) \leq .75\alpha \\ DC_{k,i} + .5\alpha - [(DC_{k,i} - .5\alpha) \bmod \alpha] + .25\alpha & Otherwise \end{cases} \quad (11)$$

where $DC_{k,i}$ and $DC'_{k,i}$ respectively are the original and embedded DC values of the i^{th} block in the k^{th} embedding unit, and α is the appropriate embedding strength for block i and is computed by (9). Here, i ranges from 1 to 4 and k ranges from 1 to the total number of embedding units. After all the watermark bits are embedded, the 8×8 block-based inverse DCT is applied to obtain the watermarked image I'.

2.5. Image Normalization

An image normalization technique [12] is finally applied to the watermarked image to make our proposed watermarking scheme further resistant to flipping attacks. Specifically, we use the ordinary geometric moments to detect the flipping attacks along x or y axis. This moment $\mu_{p,q}$ is computed as:

$$\mu_{p,q} = \iint_{\Gamma} (x - x_0)^p (y - y_0)^q f(x, y)\, dx\, dy \quad (12)$$

where (x, y) is the pixel coordinate, $f(x, y)$ is the pixel intensity at (x, y), (x_0, y_0) is the image center, and Γ is the image area. We use the first-order moments, $\mu_{1,0}$ and $\mu_{0,1}$, to counterattack the flipping along x and y axes, respectively. That is, the change in the sign of these two geometric moments is used to detect the flipping along x or y axis. For example, if the sign of $\mu_{1,0}$ is different between watermark embedding and detection, it indicates a flipping happened in x direction. Similarly, if the sign of $\mu_{0,1}$ is changed, there has been a flipping along y direction.

After calculating the geometric moments of the watermarked image, the watermark embedding procedure is successfully finished. All necessary information, such as the secret key k for generating watermark message M, feature point positions, and signs of two geometric moments, are saved for watermark detection.

3. Watermark Detection Procedure

Compared to the embedding procedure, the detection procedure should be more carefully designed. Due to possible geometric distortions, the probe image must be properly re-synchronized before watermark extraction to ensure successful detection and verification. Figure 4 shows the steps of the watermark detection procedure. These steps are described in detail in the following subsections.

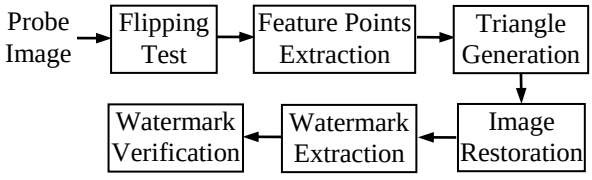


Figure 4: Watermark detection procedure.

3.1. Flipping Test

The detection procedure begins by checking the signs of two first-order geometric moments to detect possible flipping attack performed on the probe image. If the sign of $\mu_{1,0}$ or $\mu_{0,1}$ is changed according to the record, the image needs to be flipped back along either *x* or *y* axis, respectively.

3.2. Feature Points Extraction

Feature points extraction for watermark detection is exactly the same procedure as described in Section 2.1. Due to the improvement of the Harris corner detector, the number of extracted feature points is regulated by the image texture and the extracted feature points become more reliable. Consequently, the detection process uses these IFPs to perform a triangle-based self-synchronization of the image. This synchronization scheme is based on the observation that an IFP-based triangle follows the same transformation the probe image may undergo.

3.3. Triangle Generation

The triangle generation procedure plays an important role in watermark detection. Two sets of triangles are first generated by applying the Delaunay tessellation [29], an effective and repeatable method, on the IFPs extracted in the probe image and the saved IFPs, respectively. These two sets contain sufficient and useful triangles based on the IFPs and are then matched to determine the possible geometric transformations that the probe image has undergone. The determined transformations are further utilized to restore the probe image, so the synchronization errors between the extracted and original watermarks are minimized in the detection.

The choice of Delaunay tessellation is based on two attractive properties: 1) Local property: If a vertex disappears, the tessellation is only modified on connected triangles. 2) Stability area: Each vertex is associated with a stability area where the tessellation pattern is not changed when the vertex is moved within this area. That is, the tessellation patterns of other triangles remain the same even though losing or shifting an IFP affects the triangle(s) connected to it. In addition, two properties of the Delaunay tessellation always ensure that an identical generation of triangles can be obtained if the relative positions of the IFPs do not change. We implemented the Qhull algorithm [30] to generate the IFPs-based triangles due to its fast speed and less memory constraints.

3.4. Image Restoration

The image restoration procedure utilizes the triangle matching technique to determine the possible geometric transformations that the probe image has undergone and restore the probe image to be aligned with the host image. Specifically, the triangle matching technique searches all possible matched triangle pairs by comparing the angle radians. If two triangles have very similar angle radians (i.e., the angle difference is less than 0.01 radian), they are claimed to be matched. The possible geometric transformations are then determined from the matched triangle pairs since the IFPs-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. Since an image and the within triangles undergo exactly the same transformation, the majority of the identical three-element-tuples obtained from all matched triangle pairs is used to align the probe image with the host image.

It is worth mentioning that sufficient IFPs 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 3-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.

3.5. Watermark Extraction

Watermark extraction is applied after the probe image has been aligned to the original position via the image restoration. The following steps are a straightforward continuation of the process.

1. The aligned probe image is transformed into 8×8 non-overlapping DCT blocks. Every 4 spatially adjacent blocks are grouped together to form the embedding units. The watermark bit is then extracted from each of these embedding units in the same order as generated in the embedding process. That is, each of 4 DC values in every embedding unit is modularly divided by α, which is calculated using the HVS-based embedding strength calculation method described in Section 2.3. The extraction function is:
$$w'_{k,i} = \begin{cases} 1, & DC''_{k,i} \bmod \alpha \geq 0.5\alpha \\ -1, & otherwise \end{cases} \quad (13)$$
where $DC''_{k,i}$ is the i^{th} DC value in the k^{th} embedding unit, $\widetilde{w}'_{k,i}$ is one of the extracted bits in the k^{th} embedding unit, $1 \leq i \leq 4, 1 \leq k \leq N$, and *N* is the length of the watermark.
2. The final watermark bit M'_k of k^{th} embedding unit is decided by the majority value in the group $w'_{k,i}$ (i=1, 2, 3, 4).

Figure 5 illustrates the relationship between watermark embedding and extraction schemes. Based on (10) and (11), we can easily derive the following two relations: If $M_i = 1, DC'_{k,i}\ mod\ \alpha = .75\alpha$, and if $M_i = -1, DC'_{k,i}\ mod\ \alpha = .25\alpha$. As a result, the watermark detection threshold is set to be the average of 0.25α and 0.75α, i.e., 0.5α, and the maximum error tolerance of the embedded bit is $\pm 0.5\alpha$.

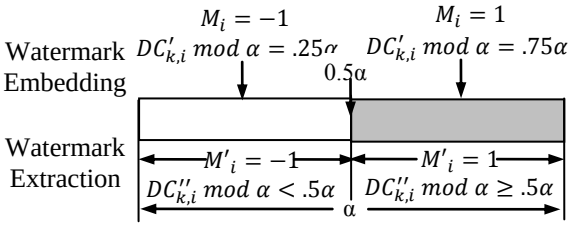


Figure 5: The relationship between the proposed watermark embedding and extraction schemes.

3.6. Watermark Verification

Watermark verification is performed using the principle of the spread spectrum method [3]. The final decision is based on the correlation value *S* between *M* and *M'*:

$$S = \frac{cov(M,M')}{\sqrt{cov(M,M)\cdot cov(M',M')}} \quad (14)$$

That is, *S* is compared with a predefined threshold *T* to decide the presence of the watermark. If *S* is larger than *T*, the watermark exists. However, a high threshold will increase the false negative rate and a low threshold will increase the false positive rate. As a result, we choose an appropriate predefined threshold *T* based on the false alarm probability that may occur in watermark detection.

The extracted watermark sequence *M'* can be considered as a normal distribution since it is a 1024-bit random sequence for an 8-bit 512×512 grayscale image. This sequence can be easily transformed into a standard normal distribution sequence, $Z = \frac{M'-\mu}{\sigma}$, assuming sequence *M'* has a mean value μ and a standard deviation σ. Based on the central limit theorem, the Cumulative Distribution Function (CDF) of a normal distribution is:

$$Prob\{x > a\} = Q\left(\frac{\alpha-\mu}{\sigma}\right) \quad (15)$$

where *Q*(*z*) is the CDF of a standard normal distribution and has the form of

$$Q(z) = \int_z^{\infty} \frac{1}{\sqrt{2\pi}} e^{\frac{-z^2}{2}}\, dx \quad (16)$$

Consequently, we choose threshold *T* by the statistical estimation of the error probability:

$$T = \beta \times \sigma + \mu$$
$$Prob\{x > T\} = Q\left(\frac{T-\mu}{\sigma}\right) \quad (17)$$

where β is the scaling parameter that is adjustable according to the requirement of the false alarm probability.

In our experiment, we run 1000 standard significance tests on the random correlation scores. In these tests, we use one of 1000 randomly generated 1024-bit sequences to correlate with all these 1000 sequences. The result is depicted in Figure 6. The strong correlation value is the result of two identical watermarks. The standard deviation σ and mean value μ are 0.038 and 0, respectively. In our system, we claim the presence of the watermark with a false alarm probability of lower than 10^{-6}. That is, $Prob\{x > T\} = Q\left(\frac{T-\mu}{\sigma}\right) \leq 10^{-6}$. By calculating Q(β), we obtain that β = 4.5 leads the CDF to be smaller than 10^{-6}. As a result, threshold *T* is set to $4.5 \times 0.038 + 0 = 0.17$ in our system to indicate the presence of the watermark.

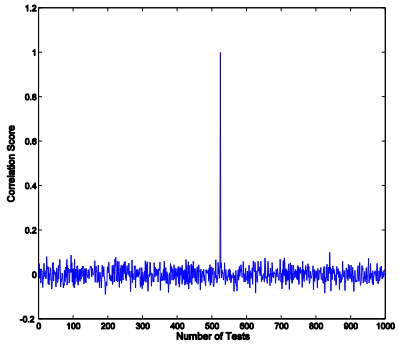


Figure 6: The result of the standard significance test.

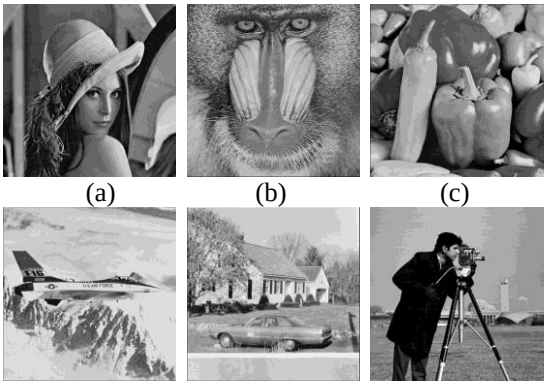
4. Experimental Results and Comparisons

In this section, we focus on a group of common operations that can be seen as attacks on watermarked images. We first perform the watermark invisibility test using six 512×512 8-bit gray-level images. We then illustrate the effectiveness of the proposed IFPs-based image restoration scheme, which functions as a self-synchronization scheme to align the possibly geometrically distorted watermarked image with the original one. Although the goal of our watermarking scheme is to be RST-resilient, it is also a very good DCT-based still image watermarking scheme. Therefore, we separate the simulation results into three parts. In the first part, we compare our scheme with two carefully selected DCT-domain schemes in terms of the resistance to common image processing. These two chosen schemes are the classical DCT algorithm proposed by Cox *et al.* [3] and a new DCT-based algorithm proposed by Suhail and Obaidat [31]. In the second part, intensive comparisons are performed with three well designed feature-based RST resilient watermarking schemes proposed by Tang and Hang [20], Wang *et al.* [21], and Bas *et al.* [16]. The experimental results will be recorded as " √" or " ×" to give an intuitive comparison instead of

using the similarity scores which are not highly significant to the comparison due to different approaches for achieving the RST resistance and different mechanisms for determining the presence of the watermark. For the ease of comparison, the results are listed side by side. That is, a method with more " √" on its side has a better performance. In the third part, we summarize the performance of our proposed system under a variety of Stirmark attacks on 105 8-bit watermarked grayscale images of size 512×512.

4.1. Watermark Invisibility Test

We evaluate watermark invisibility on six images: Lena, Baboon, Pepper, Airplane, Car, and Cameraman (Man). These six images correspond to several texture categories. For example, Baboon includes textured areas with high frequency components; Lena and Airplane include large homogeneous areas whereas Lena has sharp edges; and Pepper falls in a low-textured category. Figure 7 shows the watermarking results of all these six images. It clearly demonstrates that the watermarked images are identical to the original images to our naked eyes. The PSNR values for these six watermarked images are 41.80, 39.06, 41.16, 40.28, 37.12, and 40.98, respectively. These values are all greater than 35.00db, which is the empirical value for the image without any perceivable degradation [32].



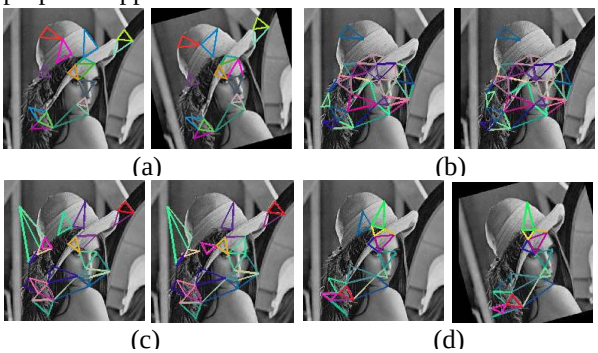
(a) (b) (c)

(d) (e) (f)

Figure 7: The visual results of images (Lena, Baboon, Pepper, Airplane, Car, and Cameraman) after being embedded with a watermark

4.2. Image Restoration Test

Image restoration is an important step in the proposed watermarking scheme. In general, we apply the Delaunay tessellation on the IFPs extracted by the image-texture-based adaptive Harris corner detector 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 attacks. Figure 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 Figure 8, we can always find sufficient matched triangles to restore the probe image. This observation further proves the effectiveness of the proposed approach.



(a) (b)

(c) (d)

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

Table 2 lists three image texture dependant parameters and the number of IFPs determined by applying our image-texture-based adaptive Harris corner detector on six test images. These parameters are *Ratio* (the factor for classifying image textures), *Type* (the texture decided by (5); i.e., H indicates high textured, M indicates medium textured, and L indicates low textured), and *D* (the diameter of the circular window used by our improved Harris corner detector). It clearly shows that diameter *D* is determined by the image texture. That is, the more complicate the texture, the larger the diameter *D*. We also observe that the number of IFPs is less than 70 for all the test images with different textures. This observation clearly demonstrates that our improved Harris corner detector does regulate the number of IFPs. It also indicates that the cost of saving IFPs for watermark synchronization is minimal compared with the cost of saving the host image. Table 2 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 Figure 8. 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 is not linearly related to the number of IFPs due to the sensitivity of the IFPs to different attacks (i.e., some IFPs may disappear, show up, or shift a bit in the attacked image). However, two properties of the Delaunay tessellation always ensure that there are

enough matched triangles, as indicated by high ratios in Table 2, for restoring the probe image.

Table 2: Three image texture dependant parameters and ratios under different attacks

Images	Image Texture Dependant Parameters			IFP	Geometric Attacks			
	Ratio	Type	D		(a)	(b)	(c)	(d)
Lena	0.002	M	42	53	14/16	35/36	10/11	14/16
Baboon	0.01	H	54	50	14/19	23/25	5/10	6/6
Pepper	0.0013	L	35	59	26/27	21/22	8/10	6/9
Plane	0.0033	M	42	46	11/15	36/37	6/8	13/15
Car	0.005	M	42	65	18/24	63/63	15/18	6/15
Man	0.0024	M	42	51	21/23	30/30	7/7	7/10

4.3 Simulation Results

4.3.1. Comparison with Basic DCT-based Watermarking Methods

Two common image processing attacks, median filtering and JPEG compression, have been extensively tested on a variety of gray-scale watermarked images. Figure 9 shows the average correlation scores of the recovered watermarks after applying median filtering of sizes from 3×3 to 8×8 on watermarked images Lena, Plane, and Cameraman generated by our proposed method, Cox's method [3], and Suhail's method [31] . The horizontal axis indicates the filter sizes, such as 3×3, 4×4, 5×5, etc. The vertical axis is the correlation value. It clearly demonstrates that the correlation scores decrease with an increase in the filter size. For example, the correlation scores of the proposed method, Suhail's method, and Cox's method are 0.98, 0.87, and 0.78 under the 3×3 median filtering attack, respectively. The three scores are 0.59, 0.57, and 0.275 under the 8×8 median filtering attack, respectively. In general, our method outperforms Suhail's and Cox's since the shape of our correlation scores is higher above those of the other two. Moreover, the proposed method performs more stable than the other two methods since it has the smoothest slope.

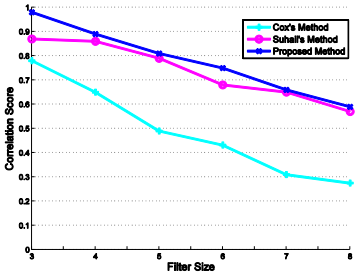


Figure 9: The average correlation scores of the recovered watermark computed by three methods after median filtering attacks.

Figure 10 shows the average correlation scores under JPEG compression attacks on watermarked images Lena, Plane, and Cameraman generated by the above three methods. The horizontal axis indicates the percentage value of the compression ratio and the vertical axis is the correlation value. It clearly shows the correlation values of the proposed method follow a smooth shape with a gradual downward movement. The correlation values of Suhail's method and Cox's method respectively decrease dramatically after 15% and 10% compression whereas our results start to drop after 25% compression. This observation means our method has a better JPEG tolerance than the other two and the effect of compression being lower than 25% is negligible to our method. In addition, the correlation values of the proposed method, Suhail's method, and Cox's method are respectively 0.76, 0.55, and 0.18 under the highest compression ratio of 45%. We also anticipate the proposed method can survive a much higher compression ratio than the other two methods based on the trend of result lines.

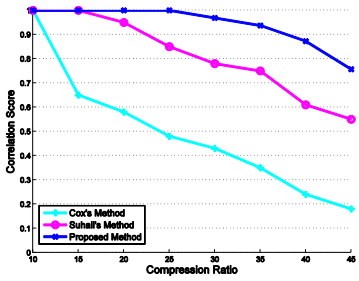


Figure 10: The average correlation scores of the recovered watermark computed by three methods after JPEG compression attacks.

In conclusion, our scheme performs well under common image processing attacks and outperforms the other two representative DCT-based digital watermarking schemes [3, 31]. This robustness against compression is owed to the DC component embedding scheme where a DC component has more embedding capacity than any of the AC components and is the most unlikely position that will be changed by an image compression algorithm.

4.3.2. Comparison with Feature-based RST Robust Watermarking Methods

The proposed method is compared with three feature-based RST robust watermarking methods, namely Tang's method [20], Wang's method [21], and Bas's method [16]. The comparison results upon various common image processing attacks and geometric distortions are listed side by side in Tables 3 through 5. A " √" means the method successfully detects the watermark after the attack and a " ×" means the method fails to detect the watermark.

Table 3 compares the proposed method with Tang's method [20], which is a feature-based DFT domain method, in terms of the robustness against different attacks such as median filtering, sharpening, Gaussian filtering, mean filtering, JPEG compression,

rows and columns removal, small shearing, rotation, scaling, translation, cropping, combined RST attacks, and flipping. All these tests are performed on images Lena (L), Baboon (B), and Pepper (P) for a fair comparison. As shown in Table 3, our scheme successfully passes all the tests while Tang's method completely fails mean filtering, rotation 2º plus cropping, rotation 5º plus cropping, scaling 90%, and scaling 80%. Our method also shows better stability under median filtering, Gaussian filtering, removal of rows or/and columns, small shearing, JPEG 40%, and combined RST attacks. Moreover, our method yields almost equal performance on all three images due to the regulation of the number of feature points on images with different textures whereas Tang's method performs better on high textured images such as Baboon. In summary, the performance of our scheme is much more stable under different attacks in the comparison with Tang's method. One reason is that our IFPs are more stable than those found by the Mexican hat detector. These robust IFPs ensure more accurate synchronization between the probe and original watermarked images. Another reason is the DC component embedding scheme embeds watermark in the most unlikely positions that will be changed by common image processing attacks.

Table 3: The comparison between the proposed method and Tang's method [20] under different image processing and geometric distortions.

Methods / Attacks	Tang's			Ours		
	L	B	P	L	B	P
No Attack	√	√	√	√	√	√
2×2 Median filtering	×	√	×	√	√	√
3×3 Median filtering	×	√	×	√	√	√
3×3 Sharpening	√	√	√	√	√	√
3×3 Gaussian Filtering	√	√	×	√	√	√
2×2 Mean Filtering	×	×	×	√	√	√
3×3 Mean Filtering	×	×	×	√	√	√
Remove 1 Row and 5 Cols.	√	√	√	√	√	√
Remove 5 Rows and 17 Cols.	×	√	×	√	√	√
Shearing *x*-1%, *y*-1%	√	√	×	√	√	√
Shearing *x*-0%, *y*-5%	√	√	×	√	√	√
Shearing *x*-5%, *y*-5%	×	√	×	√	√	√
Rotation 1º + Cropping	√	√	√	√	√	√
Rotation 2º + Cropping	×	×	×	√	√	√
Rotation 5º + Cropping	×	×	×	√	√	√
Scaling 90%	×	×	×	√	√	√
Scaling 80%	×	×	×	√	√	√
Translation [10,10]	√	√	√	√	√	√
Center Cropping 5% off	√	√	√	√	√	√
Center Cropping 10% off	√	√	√	√	√	√
Combined RST Attacks	√	√	×	√	√	√
Flipping along *x* Direction	√	√	√	√	√	√
Flipping along *y* Direction	√	√	√	√	√	√
JPEG 80%	√	√	√	√	√	√
JPEG 40%	√	√	×	√	√	√

Table 4 compares the proposed method with Wang's method [21], which is an improved feature-based DFT method, in terms of the robustness against different attacks such as median filtering, sharpening, Gaussian noise, JPEG compression, rotation, scaling, translation, cropping, local random bending, and combinations of several attacks. All these tests are performed on images Lena (L), Baboon (B), and Pepper (P) for a fair comparison. As shown in Table 4, our scheme performs better in the scaling and combinational tests. These successes are mainly due to the following reason: our proposed triangle generation and image restoration techniques ensure enough matched triangles for accurate self-synchronization under a variety of RST and aspect ratio changing attacks, while the local characteristic region derived from the scale-space theory is not geometrically invariant space, as explained in Wang's method. However, Wang's method performs better than ours in Gaussian noise attacks because we fail on Baboon image. That is because the added noise may cause a large DC magnitude change, which may directly affect the results of the watermark extraction function.

Table 4: The comparison between the proposed method and Wang's method [21] under different image processing and geometric distortions.

Methods / Attacks	Wang's			Ours		
	L	B	P	L	B	P
3×3 Median filtering	√	√	√	√	√	√
3×3 Sharpening	√	√	√	√	√	√
Gaussian Noise	√	√	√	√	×	√
JPEG 70%	√	√	√	√	√	√
JPEG 50%	√	√	√	√	√	√
JPEG 30%	√	√	√	√	√	√
3×3 Median filter + JPEG 90%	√	√	√	√	√	√
3×3 Sharpening + JPEG 90%	√	√	√	√	√	√
Remove 8 Rows and 16 Cols.	√	√	√	√	√	√
Cropping 55%	√	√	√	√	√	√
Rotation 5º	√	√	√	√	√	√
Rotation 15º	√	√	√	√	√	√
Rotation 30º	√	√	√	√	√	√
Translation [10, 10]	√	√	√	√	√	√
Scaling 0.6	×	√	√	√	√	√
Scaling 0.9	√	√	√	√	√	√
Scaling 1.4	×	×	√	√	√	√
Local random bending	√	√	√	√	√	√
Cropping 10% + JPEG 70	√	√	√	√	√	√
Rotation 5º + Scaling 0.9	√	√	√	√	√	√
Translation [10, 10] + Rotation 5º + Scaling 0.9	√	√	×	√	√	√

Table 5 compares the proposed method with Bas's method [16], which is a feature-based spatial domain method, in terms of the robustness against different attacks such as median filtering, Gaussian filtering, shearing, rotation, scaling, JPEG compression, and

Stirmark general attacks. All these tests are performed on images Lena (L), Plane (P), and Car (C) for a fair comparison. As shown in Table 5, both methods successfully pass the shearing, rotation, scaling, and Stirmark general attacks. However, our method has advantages on common image processing tests such as 8×8 median filtering and 30% JPEG compression due to the following reasons: 1) Our method embeds the watermark into the DC components which are unlikely to be changed by common image processing attacks, while Bas's method introduces an interpolation problem when doing the watermark triangle wrapping in detection. 2) Our method embeds the watermark in DCT domain, while Bas's scheme is suitable for directly adding watermarks into the spatial domain due to irregular shape of the embedding area. This makes the method vulnerable to image processing distortions.

Table 5: The comparison between the proposed method and Bas's method [16] under different image processing and geometric distortion attacks

Methods / Attacks	Bas's			Ours		
	L	P	C	L	P	C
No Attack	√	√	√	√	√	√
8×8 Median Filtering	×	×	×	√	√	√
3×3 Gaussian Filtering	√	√	√	√	√	√
Shearing *x*-1%, *y*-1%	√	√	√	√	√	√
Shearing *x*-0%, *y*-5%	√	√	√	√	√	√
Shearing *x*-5%, *y*-5%	√	√	√	√	√	√
Rotation 10º	√	√	√	√	√	√
Scaling 90%	√	√	√	√	√	√
Scaling 80%	√	√	√	√	√	√
JPEG 80%	√	√	√	√	√	√
JPEG 45%	√	√	√	√	√	√
JPEG 30%	×	×	√	√	√	√
Stirmark General Attacks	√	√	√	√	√	√

4.3.3. Comprehensive Simulation Results under Stirmark Attacks

We perform a variety of attacks on 105 8-bit watermarked grayscale images of size 512×512 using 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 40.62db. Figure 11 demonstrates the simulation results of 15 kinds of common attacks on the 105 watermarked images. The simulated attacks are listed on *x*-axis where all the numerically labeled attacks sequentially 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, 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 7×7. Each distortion category (i.e., numbers 2 to 12 on *x*-axis in Figure 11) contains 3 random attacks. The *y*-axis summarizes the average detection rates of all images in each texture level under each distortion category. Figure 11 clearly demonstrates that our scheme achieves good robustness under both image processing and geometric distortions and performs the worst for low and high textured images under the linear geometric attacks. Specifically, the average detection rates for all simulated geometric attacks are respectively 92.62%, 87.25%, and 78.21% and for all simulated image processing attacks are respectively 99.18%, 100%, and 91.35% for medium, low, and high textured images. The average detection rates for all simulated attacks are 96.61%, 94.90%, and 86.67% for medium, low, and high textured images, respectively. The overall average detection rate for all images under all simulated attacks is 92.73%.

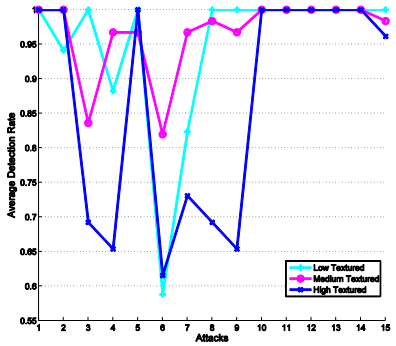


Figure 11: The average successful detection rates for three kinds of textured images under geometric and common image processing attacks using Stirmark.

In summary, the all-around result of our proposed watermark scheme outperforms the other peer schemes [3, 14, 16, 20, 21]. It yields positive detection results for most images with low, medium, and high textures under different geometric distortions and common image processing attacks. The HVS-based DCT embedding method achieves a good balance between invisibility and robustness. Furthermore, our image-texture-based adaptive Harris corner detector is capable of finding the regulated IFPs for different textured images. The triangle generation and image restoration scheme is able to efficiently minimize the synchronization errors between the extracted and original watermarks for achieving the robustness against geometric distortions. The robustness of the spread-spectrum-based DC component embedding and detection makes our scheme more resistant to common image processing attacks. The image normalization

technique is also helpful in detecting the flipping attacks. However, our method does not perform well on noise-adding tests since such attacks may cause a larger variation in DC component magnitudes than other attacks do.

5. Conclusions and Future Work

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

- Improved feature points extraction. This extraction method is capable of adaptively finding the IFPs in high, medium, and low textured images. These IFPs are more robust against geometric attacks.
- DC-components-based embedding method. This scheme achieves a higher embedding capacity and better robustness against common image processing attacks.
- HVS-based embedding strength selection. This selection integrates Watson's DCT-based visual model into our DC-components-based spread spectrum watermarking method for achieving a very good visual quality and better robustness in detection.
- Dalaunay-tessellation-based triangle generation and image restoration. This scheme can efficiently calculate the triangle tessellation in both probe and original watermarked images and accurately determine the possible transformation a probe image may undergo no matter some IFPs are lost or shifted in the probe image.

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 peer watermarking techniques. It works successfully for medium textured images and decently for low and high textured images. Our approach can be further improved by developing a more reliable feature extraction method under severe geometric distortions and a more stable DC component embedding function under noise-adding attacks.

References

[1] F. Hartung and M. Kutter. Multimedia Watermarking Techniques. *IEEE Special Issue on Identification and Protection of Multimedia Information*, 87 (7), pp. 1079-1107, 1999.

[2] C. H. Li and S. S. Wang. Transform-Based Watermarking for Digital Images and Video. *In International Conference on Consumer Electronics*, pp. 108-111, 1999.

[3] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, Secure Spread Spectrum Watermarking for Multimedia. *IEEE Transactions on Image Processing*, 6 (12), pp. 1673-1687, 1997.

[4] W. C. Cheng, C.W. Chu, and J. S. Wang. A Robust Watermarking Method on Discrete Cosine Transform Domain. *In Proceedings of the International Conference on Data Hiding and Digital Watermarking*, pp. 191-198, 1999.

[5] W. N. Lie, C. L. Wu, and J. S. Wang. Robust Image Watermarking on the DCT Domain. *In Proceedings of the IEEE Workshop on Computer Vision, Graphics and Image Processing*, pp. 9-15, 1999.

[6] J. J. K. O'Ruanaidh and T. Pun. Rotation, Scale, and Translation Invariant Digital Image Watermarking. *In Proceedings of IEEE International Conference on Image Processing*, pp. 536-539, 1997.

[7] J. J. K. O'Ruanaidh and T. Pun. Rotation, Scale, and Translation Invariant Spread Spectrum Digital Image Watermarking. *Signal Processing*, 66 (3), pp. 303-317, 1998.

[8] C. Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, M. L. Miller, and Y. M. Lui. Rotation, Scale, and Translation Resilient Watermarking for Images. *IEEE Transactions on Image Processing*, 10 (5), pp.767-782, 2001.

[9] S. Pereira, J. J. K. O'Ruanaidh,, F. Deguillaume, G. Csurka, and T. Pun. Template Based Recovery of Fourier-Based Watermarks Using Log-Polar and Log-Log Maps. *In Proceedings of IEEE International Conference on Multimedia Computing Systems*, 1, pp. 870-874, 1999.

[10] S. Pereira and T. Pun. Robust Template Matching for Affine Resistant Image Watermarks. *IEEE Transactions on Image Processing*, 9 (6), pp. 1123-1129, 2000.

[11] A. Herrigel, S. Voloshynovskiy, and Y. B. Rytsar. Watermark Template Attack. *In Proceedings of SPIE Security and Watermarking of Multimedia Contents*, 3 (4314), pp. 394-405, 2001.

[12] M. Alghoniemy and A. H. Tewfik. Geometric Distortion Correction Through Image Normalization. *In Proceedings of IEEE International Conference on Multimedia Expo*, 3, pp. 1291-1294, 2000.

[13] M. Alghoniemy and A. H. Tewfik. Image Watermarking by Moment Invariants. *In Proceedings of IEEE International Conference on Image Processing*, 2, pp. 73-76, 2000.

[14] M. Alghoniemy and A. H. Tewfik. Geometric Invariance in Image Watermarking. *IEEE Transactions on Image Processing*, 13 (2), 145-153, 2004.

[15] H. S. Kim and H. K. Lee. Invariant Image Watermark Using Zernike Moment. *IEEE Transactions on Circuit and Systems for Video Technology*, 13 (8), pp. 766-775, 2003.

[16] P. Bas, J. M. Chassery, and B. Macq. Geometrically Invariant Watermarking Using Feature Points. *IEEE Transactions on Image Processing*, 11 (9), pp. 1014-1028, 2002.

[17] J. Seo and C. Yoo. Localized Image Watermarking Based on Feature Points of Scale-Space Representation. *Pattern Recognition*, 37 (7), pp. 1365-1375, 2004.

[18] J. Seo and C. Yoo. Image Watermarking Based on Invariant Regions of Scale-Space Representation. *IEEE Transactions on Signal Processing*, 54 (4), pp. 1537-1549, 2006.

[19] H. Lee, H. Kim, and H. Lee. Robust Image Watermarking Using Local Invariant Features. *Optical Engineering*, 45 (3) pp. 1-11, 2006.

[20] C. W. Tang and H. M. Hang. A Feature-Based Robust Digital Image Watermarking Scheme. *IEEE Transactions on Signal Processing*, 51 (4), pp. 950-959, 2003.
[21] X. Wang, J. Wu, and P. Niu. A New Digital Image Watermarking Algorithm Resilient to Desynchronization Attacks. *IEEE Transactions on Information Forensics Security*, 2 (4), pp. 655-663, 2007.
[22] X. Qi and J. Qi. A Robust Content-Based Digital Image Watermarking Scheme. *Signal Processing*, 87 (6), pp. 1264-1280, 2007.
[23] C. Harris and M. Stephen. A Combined Corner and Edge Detector. *In Proceedings of Alvey Vision Conference*, pp. 147–151, 1988.
[24] C. Schmid, R. Mohr, and C. Bauckhage. Comparing and Evaluating Interest Points. *In Proc. of the 6th International Conference on Computer Vision*, pp. 230-235, 1998.
[25] Y. Q. Shi and H. Sun. *Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards*. Boca Raton, FL: CRC, 1999.
[26] J. Huang, Y. Q. Shi, and Y. Shi. Embedding Image Watermarks in DC Components. *IEEE Transactions on Circuits and Systems for Video Technology*, 10 (6), pp. 974-979, 2000.
[27] B. Watson. DCT Quantization Matrices Visually Optimized for Individual Images. *In Proceedings of SPIE: Human Vision, Visual Processing, and Digital Display IV*, 1913, pp. 202-216, 1993.
[28] H. M. Tsai, and L. W. Chang. Highly Imperceptible Video Watermarking with the Watson's DCT-Based Visual Model. *In Proceedings of IEEE International Conference on Multimedia and Expo*, pp. 971-974, 2004.
[29] E. Bertin, S. Marchand-Maillet, and J. M. Chassery. *Optimization in Voronoi Diagrams*. Kluwer, 1994.
[30] C. B. Barber, D. P. Dobkin, and H. T. Huhdanpaa. The Quickhull Algorithm for Convex Hulls, *ACM Transactions on Mathematical Software*, 22 (4), pp. 469-483, 1996.
[31] M. A. Suhail and M. S. Obaidat. Digital Watermarking-Based DCT and JPEG Model. *IEEE Transactions on Instrumentation and Measurement*, 52 (5), pp. 1640-1647, 2003.
[32] M. S. Hsieh and D. C. Tseng. Perceptual Digital Watermarking for Image Authentication in Electronic Commerce. *Electronic Commerce Research*, 4, 157-170, 2004.

Dr. Xiaojun Qi is currently an associate professor in the Department of Computer Science at Utah State University in USA. She received her M.S and Ph.D. degree in Computer Science from Louisiana State University in 1999 and 2001, respectively. Her research interests include content-based image retrieval, digital watermarking and steganography, and computer vision. She has published over 40 peer-reviewed journal and conference publications. She has been served as technical program committee for 10 international conferences. She is a member of IEEE.

Mr. Ji Qi is currently working as a Software Engineer for Electronic Arts Inc in USA. He received his M.S. degree in Computer Science from Utah State University in 2006. His research focus for his M.S. thesis is image processing and digital watermarking. He has published 6 peer-reviewed journal and conference papers while studying at Utah State University.