5-2012

A Localized Geometric-Distortion Resilient Digital Watermarking Scheme Using Two Kinds of Complementary Feature Points

Jiyuan Wang
Utah State University

A LOCALIZED GEOMETRIC-DISTORTION RESILIENT DIGITAL WATERMARKING SCHEME USING TWO KINDS OF COMPLEMENTARY FEATURE POINTS

by

Jiyuan Wang

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Computer Science

Approved:

_____	_____
Xiaojun Qi Major Professor	Stephen J. Allan Committee Member
_____	_____
Heng-Da Cheng Committee Member	Mark R. McLellan Vice President for Research Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2012

ABSTRACT

A Localized Geometric-Distortion Resilient Digital Watermarking Scheme Using Two Kinds of Complementary Feature Points

by

Jiyuan Wang, Master of Science

Utah State University, 2012

Major Professor: Dr. Xiaojun Qi
Department: Computer Science

With the rapid development of digital multimedia and internet techniques in the last few years, more and more digital images are being distributed to an ever-growing number of people for sharing, studying, or other purposes. Sharing images digitally is fast and cost-efficient thus highly desirable. However, most of those digital products are exposed without any protection. Thus, without authorization, such information can be easily transferred, copied, and tampered with by using digital multimedia editing software. Watermarking is a popular resolution to the strong need of copyright protection of digital multimedia. In the image forensics scenario, a digital watermark can be used as a tool to discriminate whether original content is tampered with or not. It is embedded on digital images as an invisible message and is used to demonstrate the proof by the owner.

In this thesis, we propose a novel localized geometric-distortion resilient digital watermarking scheme to embed two invisible messages to images. Our proposed scheme utilizes two complementary watermarking techniques, namely, local circular region (LCR)-based techniques and block discrete cosine transform (DCT)-based techniques, to

hide two pseudo-random binary sequences in two kinds of regions and extract these two sequences from their individual embedding regions. To this end, we use the histogram and mean statistically independent of the pixel position to embed one watermark in the LCRs, whose centers are the scale invariant feature transform (SIFT) feature points themselves that are robust against various affine transformations and common image processing attacks. This watermarking technique combines the advantages of SIFT feature point extraction, local histogram computing, and blind watermark embedding and extraction in the spatial domain to resist geometric distortions. We also use Watson's DCT-based visual model to embed the other watermark in several rich textured 80×80 regions not covered by any embedding LCR. This watermarking technique combines the advantages of Harris feature point extraction, triangle tessellation and matching, the human visual system (HVS), the spread spectrum-based blind watermark embedding and extraction. The proposed technique then uses these combined features in a DCT domain to resist common image processing attacks and to reduce the watermark synchronization problem at the same time.

These two techniques complement each other and therefore can resist geometric and common image processing attacks robustly. Our proposed watermarking approach is a robust watermarking technique that is capable of resisting geometric attacks, i.e., affine transformation (rotation, scaling, and translation) attacks and other common image processing (e.g., JPEG compression and filtering operations) attacks. It demonstrates more robustness and better performance as compared with some peer systems in the literature.

(62 pages)

PUBLIC ABSTRACT

A Localized Geometric Distortion Resilient Digital Watermarking Scheme Using Two Kinds of Complementary Feature Points

More and more digital images are being distributed over the Internet to an ever-growing number of people for sharing, studying, or other purposes. However, most of those digital products are exposed without any protection, and such information can be easily transferred, copied, and tampered without authorization simply by using readily available digital multimedia editing software. Digital watermarking techniques have been developed as a tool to discriminate whether the original content of digital media is tampered or not. A digital watermark is embedded on digital images as an invisible message and is used to demonstrate the proof by the owner.

In this thesis, we propose a novel localized geometric-distortion resilient digital watermarking scheme to embed two invisible messages to images. Our proposed scheme utilizes two complementary watermarking techniques, namely, local circular region (LCR)-based techniques and block discrete cosine transform (DCT)-based techniques, to hide two binary sequences in two different kinds of regions within the image and extract these two sequences from their individual embedding regions.

Working in tandem, these two methods safeguard against several common attacks to digital media. We ran several tests, the results of which demonstrate that our proposed method is more robust and has a better overall performance as compared with some peer systems in the literature.

Jiyuan Wang

ACKNOWLEDGMENTS

First I would thank my major supervisor, Dr. Xiaojun Qi. Thank you for your patience, encouragement, invaluable stimulation, and guidance throughout my research work. Thank you for providing me with constructive comments and infinite support.

I also would like to express sincere gratitude to the members of my committee, Dr. Stephen J. Allan and Dr. Heng-Da Cheng, for serving as my thesis committee members and for reviewing and giving valuable comments on my thesis.

Next, I wish to thank my schoolmates and colleagues for their support and help throughout the past few years at USU. It has been a great time to work and study with all of you.

Last and most important, I give my most sincere thanks to my parents and uncle (Charlie Wang). Their love and support will always be the strength that encourages me to pursue my goal.

Jiyuan Wang

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

BACKGROUND

In the last few years, the rapid development of digital multimedia and Internet techniques allows more and more people to enjoy the fast and convenient distribution of digital products. More and more digital images are uploaded for sharing, studying, or other purposes. However, most of digital products accessed via the Internet are without protection, and such information can be easily transferred, copied, and tampered with using digital multimedia editing software without proper authorization. Consequently, digital watermarking emerges as a possible and popular solution to resolve the strong need for protection of digital multimedia information, especially copyrighted information. Specifically, digital watermarking has been developed as a very important technology for image forensics, copyright protection, authentication, and fingerprinting. In the image forensics scenario, digital watermarking can be used as a tool to discriminate whether any original content has been tampered or not. Such watermarking 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 watermarking technique hides a message within digital media without noticeable changes to the host.

In general, watermarking techniques require several properties including transparency, robustness, trustworthy detection, and computational efficiency [1].

- Transparency means the embedded watermark should be invisible to the user. The minimum requirement of transparency is to keep the distortion

introduced by the watermark lower than the just-noticeable distortion (JND) of the image. There are different models for JND, such as contrast sensitivity function (CSF) [2] and the Watson model [3].

- Robustness is one of the most important qualities of watermarking. Basically, robustness is the watermarking technique's tolerance to common image processing methods (such as mean filtering, median filtering, and histogram equalization), geometric distortions (such as rotation, scaling, and translation), and image compression (such as JPEG compression). A robust watermark should be able to survive all those distortions.
- Trustworthy detection means the watermark detection result is able to supply a highly reliable decision as to the existence of certain watermark information. This is related to two concepts, namely, false positive and false negative. A false positive error happens in those situations in which there is no watermark in the host media, though the detector declares there is a watermark. A false negative error occurs with a negative response, even though the watermark does exist in the host media.
- Computational efficiency is the efficiency of the implementation of the watermarking scheme. That is, the watermarking procedure must be implemented in a prompt manner for its utility in the real world.

CHAPTER 2

INTRODUCTION

2.1 Digital Watermarking Procedure

Digital image watermarking imperceptibly embeds extra data into a host image. Figure 2.1 shows an overview of the watermarking process.

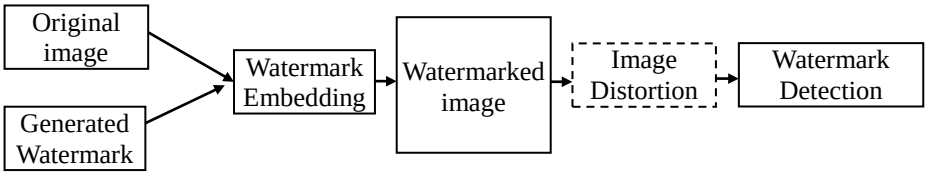


Figure 2.1. Watermarking framework.

The first step of the watermark embedding procedure is to generate the unique watermark by a secret key, which is different from all others. A common watermark is either a binary pseudo-random sequence or a binary image.

The second step is to embed the watermark. There are a variety of watermark embedding methods proposed in the literature. These techniques usually embed watermarks in either a spatial domain or a frequency domain. As a result, basic watermarking techniques can be roughly divided into two categories: spatial domain-based and frequency domain-based.

Early watermarking techniques directly embed watermark into the image (the spatial domain) by interpolating the intensity value of the original pixels in the image. These spatial domain-based watermarking embedding techniques can embed relatively large amounts of data into the image. However, they generally are not robust to image distortions. Consequently, recent watermarking techniques do not directly change the

pixel values in the image. Instead, they first transform the image into another frequency domain by applying any of several common transforms such as discrete cosine transform (DCT), discrete Fourier transform (DFT), or discrete wavelet transform (DWT), on the original image. They then embed the watermark in the newly transformed domain. These frequency domain-based watermarking techniques offer better robustness to distortion attacks than spatial domain-based watermarking techniques. In addition, they offer two more desired properties, namely, higher invisibility and stronger compression resistance. Consequently, thery are most often used by modern watermarking techniques.

The watermarked image may go through certain intentional or un-intentional distortions in the real world. As a result, a watermark detection scheme should be robust in finding and verifying the embedded watermark under possible distortions. The robustness of the watermark to common image processing and geometric attacks is important to the copyright marking system [4]. Some simple methods are presented in [4] for hiding a watermark message. However, said methods are not robust to geometric distortions. Geometric distortions are very difficult to tackle because they can make the verification task unreliable by inducing synchronization errors between the extracted and original watermarking positions during the detection process. Several state-of-the-art watermarking schemes have been developed to counterattack geometric distortions. These geometric-resilient schemes can be roughly classified into four categories: exhaustive search-based, invariant domain-based, template-based, and feature-based.

2.2. Second-Generation Geometric Resilient Watermarking Techniques

We briefly review geometric-distortion resilient watermarking techniques in each of four categories. Exhaustive search-based watermarking techniques exhaustively

search for watermarks in a large search space. They have a high computational cost and therefore cannot be effectively used in real-world applications. Invariant domain-based watermarking techniques generally provide a rotation, scaling, and translation (RST) invariant domain for embedding watermarks and maintain synchronization under affine transforms. However, they are susceptible to interpolation accuracy issues, implementation issues, and are vulnerable to cropping. Template-based watermarking techniques embed templates to identify the geometric transformation and assist watermark synchronization in the detection process. However, they usually suffer from both template estimation attacks and cropping attacks. By contrast, feature-based watermarking techniques use image dependent features as a content descriptor to represent invariant reference points for both embedding and detection. They are resistant to various attacks including cropping and random binding attacks (RBA) by binding the watermark synchronization with the image salient characteristics. These characteristics may be the whole image, some local region or regions, or feature points. This class of watermark synchronization techniques, also known as second-generation watermarking [5], has the highly desirable properties of invariance to noise, covariance to geometrical transformations and localization.

Second-generation watermarking can be divided into three sub-categories: moment-based, histogram-based, and feature point-based. In the following, we review techniques in each sub-category since our proposed system uses feature-based watermarking techniques.

2.2.1 Moment-Based Watermarking Techniques

Moment-based watermarking techniques utilize moments to solve the geometric

invariance problem. Due to their ability of representing global features, moments have been used in many applications in the field of image processing. Geometric moments are mainly used to capture global features of images. In [6–8], the watermark is embedded into a moment-based normalized image to resist affine transformation. In [9-10], Zernike moments are used as geometrically robust image watermarks. Zhang et al. [11] propose a geometric invariant blind image watermarking by using invariant Tchebichef moments and independent component analysis (ICA). However, moment-based methods are highly vulnerable to cropping.

2.2.2 Histogram-Based Watermarking Techniques

Histogram-based watermarking techniques utilize histograms to solve the geometric invariance problem. A histogram measures the global features of all pixels in an image. The histogram distribution of an image is approximately invariant under geometric attacks. For this reason, some histogram-based watermarking schemes have been presented for the purpose of robust watermarking. Xiang et al. [12] propose an invariant image watermarking in the low-frequency domain by using the histogram shape and mean in the Gaussian filtered low-frequency component of images. Coltuc and Bolon [13] propose a histogram specification-based robust watermarking scheme to embed watermarks in images. A class of watermarks is selected such that the presence of certain groups of consecutive gray levels is considerably reduced with no visual degradation of images. Chareyron et al. [14] apply the histogram specification method to chromatic histograms and color histograms based on segmentation of the XYZ color space for embedding watermark in color images. Lin et al. [15] present a histogram-oriented blind watermarking algorithm based on the three-dimensional color histogram to

resist geometric attacks and common image processing operations. The major limitation of these methods is their incapacity to resist local transformations. As a result, Deng et al. [16] developed a geometrically robust image watermarking scheme by using a histogram in a certain range to embed a watermark in circular regions centered on the Harris-Laplace feature points.

2.2.3. Feature Point-Based Watermarking Techniques

Feature point-based watermarking techniques use feature points to form local regions for embedding and extracting watermark. Lowe [17] presents a scale invariant feature transform (SIFT) detector as a feature point detector. It has been proven to be invariant to image rotation, scaling, translation, partial illumination changes, and projective transformations. This feature detector has been widely used in digital watermarking schemes to extract features. For example, Li et al. [18] embed a binary watermark image into multi-scale SIFT feature point-based local characteristic regions in the transform domain to achieve high capacity information hiding and generalized watermark robustness. Seo and Yoo [19] use the synchronization of the Harris-Laplacian feature points to achieve resilience against geometric distortions. Specifically, they embed a watermark in circularly symmetric way centered at each selected feature point. Tang and Hang [20] apply the Mexican Hat wavelet scale interaction technique to extract feature points in their proposed feature point-based robust watermark with image normalization. The image normalization technique developed for pattern recognition [21] is used for digital watermarking. Bas et al. [22] present a robust watermarking scheme based on Harris feature points. The authors apply Delaunay tessellation on the extracted Harris feature points to obtain a set of unique triangles and embed and extract a

watermark in the warped right triangles.

CHAPTER 3

THE PROPOSED GEOMETRIC-RESILIENT WATERMARKING SCHEME

In this chapter, we present our proposed geometric resilient watermarking scheme in detail. In general, our scheme belongs to the second generation watermarking methods (e.g., feature-based watermarking algorithms). It is a robust watermarking technique capable of resisting geometric attacks, i.e., affine transformation (rotation, scaling, and translation) attacks and other common image processing (e.g., JPEG compression and filtering operations) attacks. Specifically, we use two complementary watermarking techniques to hide two pseudo-random binary sequences in two kinds of regions and extract these two sequences from their individual embedding regions. To this end, we use the histogram and mean statistically independent of the pixel position to embed one watermark in the local circular regions (LCRs), whose centers are the SIFT feature points themselves and are robust against various affine transformations and common image processing attacks. This watermarking technique combines the advantages of SIFT feature point extraction, local histogram computing, and blind watermark embedding and extraction in the spatial domain to resist geometric distortions. We also use Watson's DCT-based visual model to embed other watermarks in several rich textured 80×80 regions not covered by any embedding LCR. This watermarking technique combines the advantage of Harris feature point extraction, triangle tessellation and matching, the human visual system (HVS), the spread spectrum-based blind watermark embedding and extraction in a DCT domain to resist common image processing attacks and to reduce the watermark synchronization problem at the same time. These two techniques complement

with each other, making them more resistant to geometric and common image processing attacks.

3.1 Watermark Embedding Procedure

Figure 3.1 shows the proposed watermark embedding procedure, which contains two complementary embedding techniques: LCR-based embedding and block DCT-based embedding techniques. We use a secret private key *pk* to generate two watermarks of different lengths. This key is kept by the owner to make sure the two watermarks are secure. First, we generate a 20-bit pseudo-random bipolar (e.g., 0 and 1) sequence to be embedded into two 20-bin histograms in each chosen LCR. We set the length of the watermark to be 20 since our extensive experiments show that setting the bin number to 20 generally produces a sufficient number of good quality bins in a local histogram of LCR for both embedding and detection procedures. Second, we generate a 25-bit

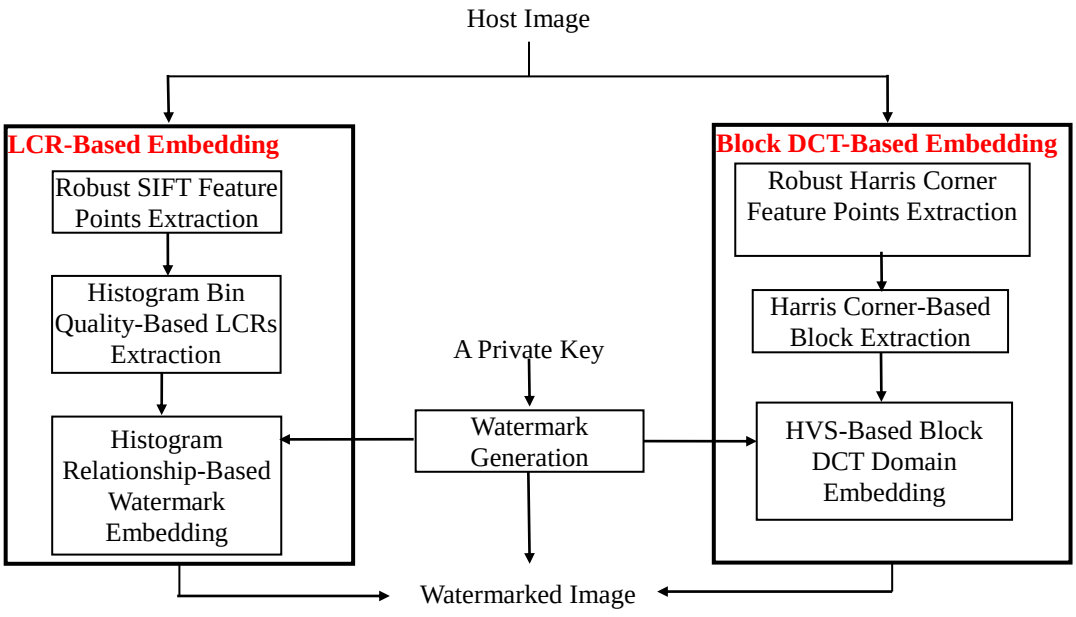


Figure 3.1. Watermark embedding procedure.

pseudo-random sequence (e.g., 1 and -1) to be embedded into the rich textured 80×80 square regions outside of any embedded LCR. We set the length of the watermark to be 25 since it is the maximum allowed payload for the 80×80 square region based on the block DCT-based embedding scheme.

3.1.1 LCR-Based Embedding Technique

The LCR-based embedding technique consists of the following three steps:

1) It applies several combined pre-attacks on SIFT feature points in a certain robust scale range to find robust SIFT feature points.

2) It divides each LCR, whose center is one of the robust SIFT feature points, into two concentric circles to split the local histogram bins and uses a histogram bin quality-based strategy to choose the best non-overlapping LCRs for embedding watermark.

3) It uses the histogram and mean statistically independent of the pixel position to embed watermark in each LCR.

The splitting strategy together with the histogram bin quality-based strategy make the proposed system easier to embed and more robust against RST attacks.

3.1.1.1 Robust SIFT Feature Points Extraction. Feature points extraction is important in the proposed digital image watermarking scheme. Feature points should be very robust and resistant to various types of geometric attacks so that watermarks can be detected without saving any information from the original images. In other words, we look for important image content-based points that are perceptually significant and can resist various types of common image processing and geometric distortions.

We tested several popular feature points extraction packages, including SIFT,

SURF [23], and Harris-Laplace [16]. We found that SIFT is more stable and robust to extract feature points. As a result, we use SIFT package to extract feature points in our system. These feature points are detected from the scale space of the image [17]. Given a digital image $I(x, y)$, its scale space representation, $L(x, y, \sigma)$, can be obtained as follows:

$$L(x, y, \sigma) = I(x, y) * G(x, y, \sigma) \quad (1)$$

where * is the convolution operator, $G(x, y, \sigma)$ is the variable-scale Gaussian kernel with standard deviation σ. The initial SIFT feature points can be detected by finding the scale space extrema in the difference-of-Gaussian (DoG) function, $D(x, y, \sigma)$, which can be obtained by subtracting two nearby scales separated by a constant multiplicative factor m:

$$D(x, y, \sigma) = L(x, y, m\sigma) - L(x, y, \sigma) \quad (2)$$

The feature points that have low contrast or are poorly localized along edges are removed. Each remaining feature point is assigned a constant orientation based on the local image properties. A highly distinctive descriptor can also be computed for each feature point for reliable image matching. Each feature point can then be represented by a vector containing the following information: x-coordinate, y-coordinate, characteristic scale σ, orientation θ, and the distinctive descriptor.

However, the SIFT package usually extracts over 1000 feature points for a grayscale image of size 512 ×512. Not all of these features points are robust against geometric attacks. We apply a series of operations to remove a significantly large number of non-robust feature points. First, we remove the relatively non-robust feature points whose scales are smaller than 4 or larger than 8 since these feature points are sensitive to scaling and rotation attacks. Second, we pre-attack the original image by

performing a combined rotation, scaling, and JPEG compression attack. Specifically, we use the combination of a rotation angle of 5˚ to 30˚ with the step size of 5˚, a scaling factor of 0.9 to 1.1 with the step size of 0.1, and a JPEG compression factor of 100 down to 70 with a step size of 10 to individually pre-attack the original image. For each pre-attacked image, we find the matched relatively robust feature points between the original image and the pre-attacked image. The intersection of these matched feature points across all the pre-attacked images and the original image keeps the robust feature points. Third, we remove the non-robust feature points that are near the image border. To this end, we remove the robust feature points whose horizontal or vertical distance to the image border is less than a constant (e.g., 8) multiplying their scale σ's. In other words, we remove robust feature points that cannot form a complete LCR for embedding watermark.

3.1.1.2 Histogram Bin Quality-Based LCRs Extraction. LCRs are the circular regions centering on the feature points. As a result, there is a LCR for each robust feature point extracted in the previous step. The radius of the LCR depends on the scale σ of its feature point, which is the center of the LCR. In our system, we empirically set the radius of each LCR as follows:

$$r = \tau \cdot [\sigma] \tag{3}$$

where $[\cdot]$ is a rounding operation and τ is a positive integer, which is used to adjust the size of a LCR. We empirically set τ to be 8.

However, LCRs may overlap if their feature points are close and their radii are large. Our extensive experiments show different selections of non-overlapping LCRs significantly affect performance. To solve this problem, we design a histogram bin quality-based strategy to remove overlapping LCRs. To this end, we first split each LCR

into two concentric circles as shown in Figure 3.2, where C_1 represents the area of the outer circular ring and C_2 represents the area of the inner circle, and where the areas of C1 and C2 are equal. We then compute a local histogram for two areas C_1 and C_2. The histogram with equal-sized bins is described as follows:

$$H = \{h(i) \mid i = 1, \cdots, L_h\} \tag{4}$$

where H is a vector denoting the gray-level histogram of an image, $h(i)$ is the number of pixels in the ith bin, and L_h is the total number of bins and set to be 20. In our system, we compute the histogram of the pixels falling in the range of B since we exclusively embed watermark in the pixel intensities in this range. Here, $B = [(1-\lambda)\overline{A}, (1+\lambda)\overline{A}]$ where $\overline{A}$ is the average intensity value of the LCR, and λ is a positive number and controls the histogram width and the quality of the watermarked image. It should be noted that a large value of λ decreases the image quality and makes the detection of watermarks more robust. Similarly, a small value of λ increases the image quality and makes the detection of watermarks difficult due to small changes. As a result, the value of λ should be wisely

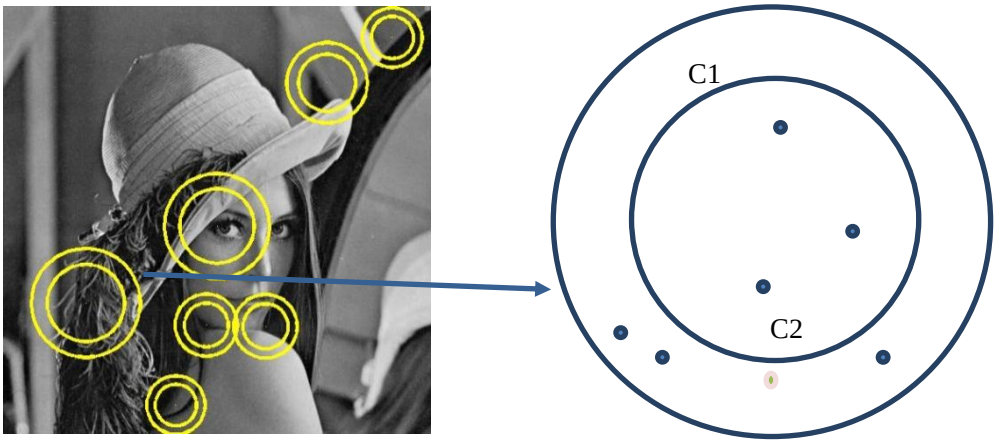


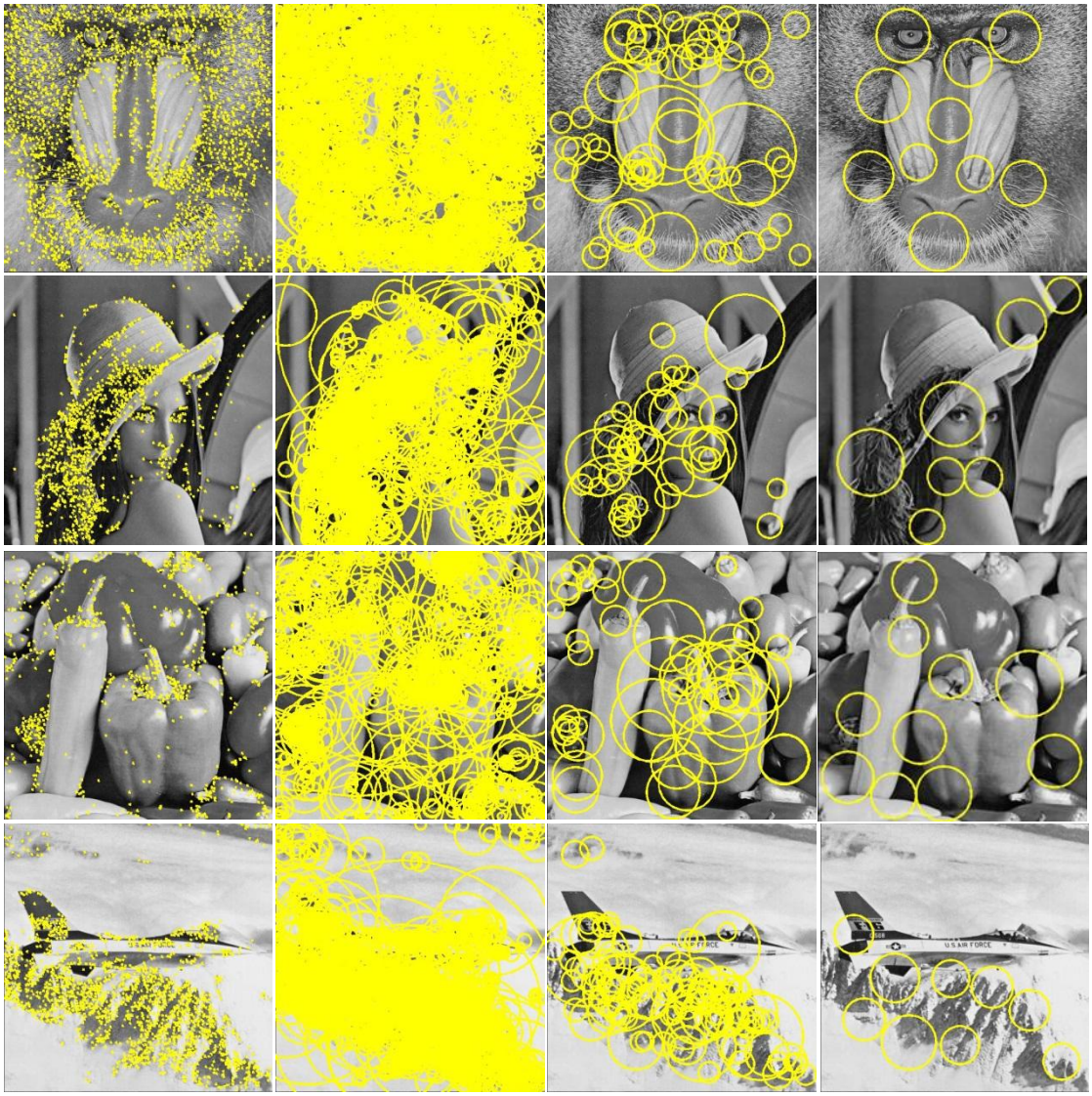
Figure 3.2. Illustration of splitting one LCR into two concentric circles.

chosen to compromise the invisibility and the robustness. In our system, we empirically set λ to be 0.6. The width of each histogram bin, *M*, is computed by:

$$\frac{(1+\lambda)\overline{A}-(1-\lambda)\overline{A}}{L_h} \tag{5}$$

After computing the local histogram for C_1 and C_2 using (4) and (5), we sort all LCRs based on the number of good quality bins in a descending order. Here, we define a good quality bin as a bin containing a sufficient number of pixels (e.g., more than 80 pixels). We then sort on the previously sorted LCRs based on the total number of pixels in all bins in a descending order. In other words, the LCR with all bins as good quality bins and the maximum pixels in all bins is the best LCR for embedding a watermark and therefore is ranked as the first in the sorted LCRs. We select this LCR at first. We then find the second best LCR that does not overlap with the best LCR. The same process is iteratively used to find all the other LCRs to be used for embedding watermark. Figure 3.2 shows final seven non-overlapping LCRs that are used for embedding the watermark. It also shows the concentric circles for each LCR and a blow-up of one sample concentric circle with a few pixels whose intensities are in the range of *B*. The watermark only changes the pixels in the range of *B* using the embedding rule explained in Section 3.1.1.3.

Figure 3.3 shows several important intermediate results for four sample images, namely, Baboon, Lena, Pepper, and Airplane. It clearly shows that the number of robust SIFT feature points is a small portion of the SIFT feature points. Keeping these robust SIFT feature points significantly reduces the computational cost in both watermark embedding and detection procedures. The number of final non-overlapping LCRs for



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

Figure 3.3. Illustration of SIFT feature points and LCR extraction. (a) SIFT feature points of original image, whose scales are in the range of [3, 7]. (b) LCRs whose centers are the SIFT feature points shown in (a). (c) LCRs whose centers are the robust SIFT feature points found by applying several pre-attacks. (d) Non-overlapping LCRs obtained by histogram bin quality-based LCR removal strategy. These non-overlapping LCRs are used to embed the 20-bit watermark.

these four sample images is 10, 7, 10, and 9, respectively. These non-overlapping LCRs will be used for embedding watermark.

3.1.1.3 Histogram Relationship-Based Watermark Embedding. We utilize the relative relationship between groups of two adjacent bins in C_1 and C_2 to sequentially embed a watermark bit. To ease the discussion, we define several notations:

- HC_1: The L_h-bin histogram in C_1 area;
- $HC_1(i)$: The i^{th} bin of HC_1;
- $HC_1(i+1)$: The $i+1^{th}$ bin of HC_1;
- a_i: The number of pixels in $HC_1(i)$;
- a_{i+1}: The number of pixels in $HC_1(i+1)$.

We sequentially choose two consecutive bins in HC_1, e.g., $HC_1(i)$ and $HC_1(i+1)$, to embed a watermark bit. The basic embedding idea is to ensure that a larger ratio of a_i and a_{i+1} is present after embedding a watermark bit of 1 and a larger ratio of a_{i+1} and a_i is present after embedding a watermark bit of 0. The detailed embedding strategy is summarized below where T is a threshold for the ratio of the pixel counts of two consecutive bins, which controls the quality of the watermarked image.

- If the embedded watermark bit is 1 and $a_i / a_{i+1} \geq T$, no operation is performed.
- If the embedded watermark bit is 1 and $a_i / a_{i+1} < T$, randomly select I_1 pixels from $HC_1(i+1)$ and subtract these chosen pixel intensities by the width of histogram bin, M. Here, I_1 is computed by:

$$I_1 = \frac{T \times a_{i+1} - a_i}{1+T} \tag{6}$$

This operation is equivalent to moving I_1 pixels from $HC_1(i+1)$ to $HC_1(i)$ to

achieve $a_i'/a_{i+1}' \geq T$, where a_i' and a_{i+1}' are the number of pixels in the two modified consecutive bins $HC_1(i)$ and $HC_1(i+1)$, respectively.

- If the embedded watermark bit is 0 and $a_{i+1}/a_i \geq T$, no operation is performed.
- If the embedded watermark is 0 and $a_{i+1}/a_i < T$, randomly select I_0 pixels from $HC_1(i)$ and add these chosen pixel intensities by the width of histogram bin, M. Here, I_0 is computed by:

$$I_0 = \frac{T \times a_i - a_{i+1}}{1+T} \tag{7}$$

This operation is equivalent to moving I_0 pixels from $HC_1(i)$ to $HC_1(i+1)$ to achieve $a_{i+1}'/a_i' \geq T$, where a_i' and a_{i+1}' are the number of pixels in the two modified consecutive bins $HC_1(i)$ and $HC_1(i+1)$, respectively.

The same embedding strategy is applied on the histogram bins in C_2 area to embed the remaining half of the watermark bits.

It should be noted that the choice of the threshold T is important. For example, the smaller T value leads to smaller changes in the watermarked image and less robustness to the attacks. The larger T value leads to bigger changes in the watermarked image and more robustness to the attacks. In our system, we set the value of T as 5, which achieves a good compromise between image quality and robustness.

3.1.2 Block DCT-Based Embedding Technique

The block DCT-based embedding technique consists of the following three steps:

1) It uses Qi and Qi's improved Harris corner detector [24] to find several robust Harris corner feature points that show different properties as the SIFT robust feature points.

2) It divides the original image into 80×80 non-overlapping blocks and locates the candidate blocks for embedding watermark using the number of robust Harris corner feature points.

3) It further divides each candidate block into 8×8 non-overlapping sub-blocks and embeds the watermark in the DC components of each sub-block using its HVS-based embedding strength.

3.1.2.1 Robust Harris Corner Feature Points Extraction. Harris corner detector is the most stable with regards to the property of repeatability under different distorted versions of the same scene. To obtain a relatively small number of robust feature points that are complementary to the SIFT feature points, we apply Qi and Qi's improved Harris corner detector [24] to find the important and robust Harris feature points. We also save the locations of these robust feature points for restoring an image in the detection procedure.

Harris and Stephen [25] improve the Harris corner detection function by using the following shape-factor-based matrix:

$$M(x,y)=\begin{bmatrix} A_{x,y} & C_{x,y} \\ C_{x,y} & B_{x,y} \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial I(x,y)}{\partial x}\right)^2 & \left(\frac{\partial I(x,y)}{\partial x}\right)\left(\frac{\partial I(x,y)}{\partial y}\right) \\ \left(\frac{\partial I(x,y)}{\partial x}\right)\left(\frac{\partial I(x,y)}{\partial y}\right) & \left(\frac{\partial I(x,y)}{\partial y}\right)^2 \end{bmatrix} \quad (8)$$

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

corner response function *R(x,y)*:

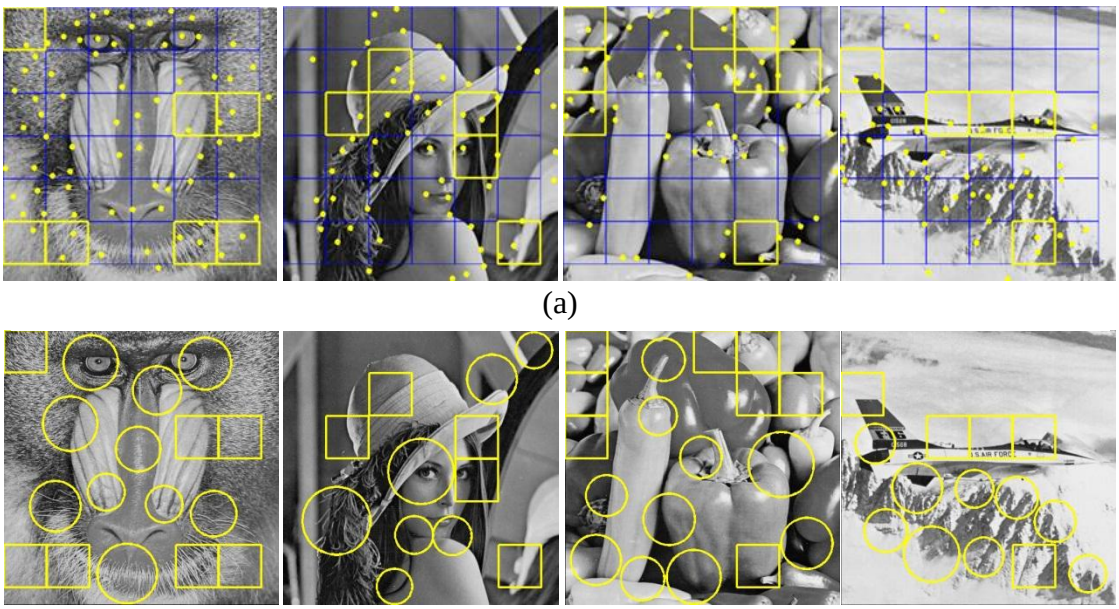
$$\begin{aligned} R(x,y) &= \det(M(x,y)) - k[trace(M(x,y))]^2 \\ &= \left(A_{x,y}B_{x,y} - C_{x,y}^2\right) - k\left(A_{x,y} + B_{x,y}\right)^2 \end{aligned} \tag{9}$$

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

Qi and Qi's improved Harris corner detector [24] further applies some pre-processing techniques to reduce the noise effect and regulate the number of important feature points based on the texture of the image.

3.1.2.2 Harris Corner-Based Block Extraction. We divide the original image into 80×80 non-overlapping blocks. We perform two filtering operations to find all candidate blocks for embedding the second watermark. We first find blocks that do not overlap with any of the embedding LCRs. We then keep such blocks that contain at least one robust Harris feature point. The resultant blocks are the candidate blocks for embedding the second watermark. Since all these candidate blocks contain at least one robust Harris feature point, they are highly textured regions suitable for embedding a watermark without causing any visual distortions. Figure 3.4 shows the robust Harris corner feature points together with the candidate blocks marked by yellow borders. The other blue bordered blocks are not used for embedding the second watermark since they contain no robust Harris corner feature points nor do they overlap with the embedding LCRs. The number of embedding blocks for Baboon, Lena, Pepper, and Airplane is 7, 5, 8, and 5, respectively.

3.1.2.3 HVS-based Block DCT Domain Embedding. For each candidate block, we further divide it into non-overlapping sub-blocks of size 8×8. Each sub-block is separately transformed by the DCT to form a DCT domain sub-block. This is consistent with the JPEG standard. We then use Watson's DCT-based visual model as the



(a)

(b)

Figure 3.4. Illustration of Harris corner feature points-based 80×80 embedding blocks and the SIFT feature points-based embedding LCRs. (a) Harris corner feature points-based embedding blocks. (b) Harris corner feature points-based embedding blocks and non-overlapping SIFT feature points-based embedding LCRs.

HVS model [25], which estimates the sensitivity of human eyes to the changes in each DCT domain sub-block. Specifically, we compute a quantitative measure of the embedding capacity of each DCT domain sub-block using the luminance and contrast masks.

The luminance masked threshold for each 8×8 sub-block is defined as:

$$t_L[i,j,k] = t[i,j]\left(C_0[0,0,k]/C_{0,0}\right)^{ar}$$
$$0 \le i,j \le 7,\ 0 \le k \le N-1 \tag{10}$$

where ar is a constant with an empirical value of 0.649, $C_0[0,0,k]$ represents the DC coefficient of the k^{th} sub-block in the candidate block, $t[i,j]$ is the DCT frequency sensitivity as shown in Table 3.1, $C_{0,0}$ is the average value of the DC coefficients in the candidate block.

Table 3.1. DCT Frequency Sensitivity Table.

1.40	1.01	1.16	1.66	2.40	3.43	4.79	5.55
1.01	1.45	1.32	1.52	2.00	2.71	3.67	4.93
1.16	1.32	2.24	2.59	2.98	3.64	4.60	5.88
1.65	1.52	2.56	3.77	4.55	5.30	6.28	7.60
2.40	2.00	2.98	4.55	6.15	7.46	8.71	10.17
3.43	2.71	3.64	5.30	7.46	9.62	11.58	13.51
4.79	3.67	4.60	6.28	8.71	11.58	14.50	17.29
6.56	4.93	5.88	7.60	10.17	13.51	17.29	21.15

The contrast masked threshold of each DCT frequency in the sub-block is calculated by:

$$s[i, j, k] = \max\left\{t_L[i, j, k], |C_0[i, j, k]|^{0.7} \times t_L[i, j, k]^{0.3}\right\} \qquad (11)$$

where, $t_L[i, j, k]$ is the luminance masked threshold for each DCT frequency in the *k*th 8×8 sub-block, and $C_0[i, j, k]$ is the DCT coefficient in *k*th 8×8 sub-block. In Watson's model, the contrast threshold value depends on both the energy present in that frequency and the luminance masked threshold for that frequency. The final result $s[i, j, k]$ is an estimation of the amounts by which individual terms of the sub-block DCT may be changed before exceeding the just noticeable distortion (JND).

The capacity of a sub-block is defined as the summation of the contrast masked threshold in the candidate block. It is computed by:

$$S_k = \sum_{i=1}^{8}\sum_{j=1}^{8} s[i, j, k] \qquad (12)$$

where $s[i, j, k]$ is the $(i, j)^{th}$ contrast masked threshold of the *k*th DCT sub-block.

We decide the embedding strength α for each DCT sub-block *k* based on its capacity. If the capacity of the *k*th sub-block is larger than the average of the mean and maximum capacities among all 100 sub-blocks in the candidate block, we set its embedding strength α as 90. Otherwise, we set its embedding strength α as 45. Our extensive experimental results show that the embedding strength of 45 can always achieve good invisibility in all the embedding areas, so we choose this value for low capacity sub-blocks.

Figure 3.5 shows the proposed strategy for generating embedding positions. That is, every 4 adjacent 8×8 sub-blocks are grouped together and embedded with a single watermark bit to increase the redundancy of the embedded information. Each of these four sub-block groups is called one embedding unit. For example, the group of sub-blocks A, B, C, and D is an embedding unit. Since each candidate block size is 80×80, the maximum number of embedding units is 25. This also means the maximum length of

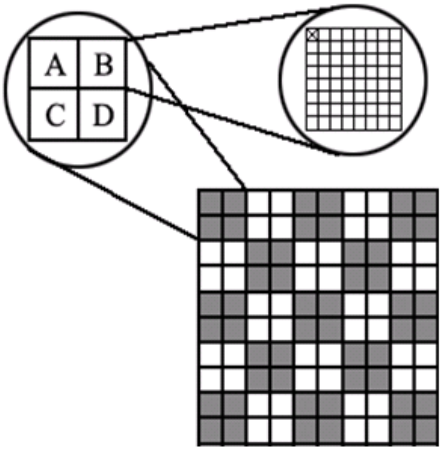


Figure 3.5. Example of embedding units.

the watermark bit sequence for the HVS-based DCT block embedding technique is 25. The embedding positions in each embedding unit are the DC components (i.e., the left top value of each 8×8 DCT sub-block shown as a check mark in Figure 3.5) of the four DCT sub-blocks. After the embedding positions are selected, the watermarked DC values, $DC_{k,i}'$, are used to replace the original DC values, $DC_{k,i}$. The *ith* DC value of the kth embedding unit, $DC_{k,i}'$, is calculated by (13), where α is the watermark embedding strength determined by HVS, and *N* is the length of the watermark.

$$\begin{aligned}
&DC_{k,i}' = DC_{k,i} - (DC_{k,i} \bmod \alpha) + \frac{3}{4}\alpha, && \text{if } w_k = 1 \text{ and } (DC_{k,i} \bmod \alpha) \geq \frac{1}{4}\alpha \\
&DC_{k,i}' = \left[DC_{k,i} - \frac{1}{4}\alpha \right] - \left[\left(DC_{k,i} - \frac{1}{4}\alpha \right) \bmod \alpha \right] + \frac{3}{4}\alpha, && \text{if } w_i = 1 \text{ and } (DC_{k,i} \bmod \alpha) < \frac{1}{4}\alpha \\
&DC_{k,i}' = DC_{k,i} - (DC_{k,i} \bmod \alpha) + \frac{1}{4}\alpha, && \text{if } w_i = -1 \text{ and } (DC_{k,i} \bmod \alpha) \leq \frac{3}{4}\alpha \\
&DC_{k,i}' = \left[DC_{k,i} + \frac{1}{2}\alpha \right] - \left[\left(DC_{k,i} - \frac{1}{2}\alpha \right) \bmod \alpha \right] + \frac{1}{4}\alpha, && \text{if } w_i = -1 \text{ and } (DC_{k,i} \bmod \alpha) > \frac{3}{4}\alpha \\
& && i = 1...4;\ k = 1...N
\end{aligned} \tag{13}$$

After embedding the 25-bit watermark sequence, we transform the modified DCT block back to the spatial domain to get the watermarked portion for HVS-based block DCT domain embedding.

3.2 Watermark Detection Procedure

Compared to the watermark 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 3.6 shows the block diagram of the watermark detection procedure. It contains two complementary watermarking detection techniques: LCR-based detection and block DCT-based detection techniques. We use the same secret

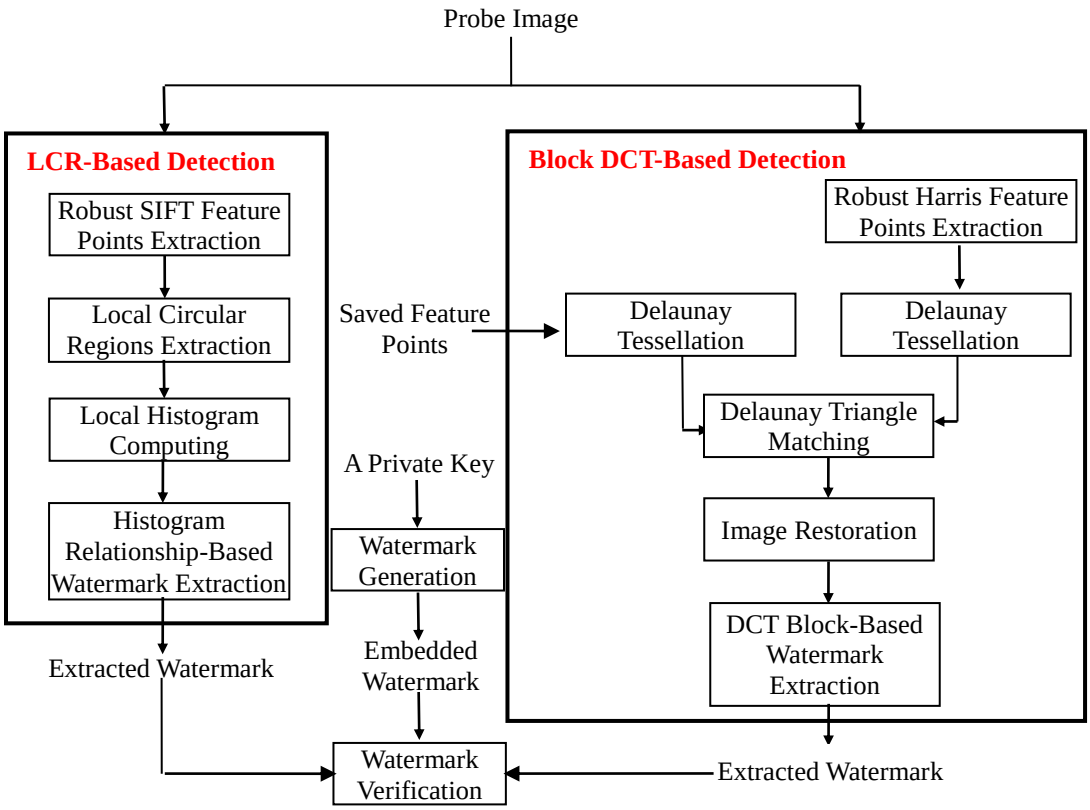


Figure 3.6. Watermark detection procedure.

private key *pk* to generate two watermarks of different lengths.

3.2.1 LCR-Based Watermark Detection

The LCR-based watermark detection technique first extracts robust SIFT feature points whose scale is in the range of 3.5 to 10. It should be noted that this is a larger scale range than the one used in an embedding procedure. This larger scale range ensures most if not all feature points used in the embedding procedure are located after any possible geometric or common image processing attacks. It then applies (3) to compute the radius of each filtered LCR and splits each LCR into two concentric circles as shown in Figure 3.2. It finally applies (4) and (5) to compute the L_h-bin local histogram in the range of B in C_1 and C_2, where L_h=20 and $B = [(1-\lambda)\bar{A}, (1+\lambda)\bar{A}]$ with $\bar{A}$ being the

average intensity value of the LCR and λ being is a positive number (e.g., 0.6). Let a_i' and a_{i+1}' be the number of pixels in two adjacent bins in C_1 or C_2. The watermark is sequentially extracted from each pair of adjacent bins in C_1 and C_2 using the histogram relationship as follows:

$$w' = \begin{cases} 1 & if \ \ a_{i+1}'/a_i' \geq 1 \\ 0 & otherwise \end{cases} \tag{14}$$

Finally, it applies the watermark verification technique to decide the presence of the watermark. Specifically, the extracted watermark sequence is compared with the secret key generated embedded watermark sequence. A ratio of matched watermark bits and the total number of watermark bits is computed for each probe LCR. We consider LCRs with a ratio of larger than 0.84 (i.e., at most a three-bit difference) containing a watermark. If at least three LCRs contain a watermark, we claim that the presence of watermark in the probe image.

It should be noted that we exclusively search all LCRs centering on the robust SIFT feature points in the detection procedure. It is possible that the final LCRs with a ratio of larger than 0.85 may overlap to a significant level. To this end, we only keep one LCR whose histogram bin quality is the best when the overlapping level is larger than 80% of the larger LCR.

3.2.2 Block DCT-Based Watermark Detection

Block DCT-based watermark detection first extracts robust Harris corner feature points as did in the embedding process [26]. Second, it applies the Delaunay tessellation on the extracted robust Harris corner feature points to generate a set of unique, non-overlapping triangles. We use the Delaunay tessellation due to its attractive properties as follows:

- *Local property*: If a vertex disappears, the tessellation is only modified on 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 [27].

Third, it applies the Delaunay tessellation on the stored robust Harris corner feature points of the original image to generate another set of unique, non-overlapping triangles. Fourth, it performs Delaunay triangle matching on the two sets of triangles to find all matched triangles. The triangle-based 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 radian), these two triangles are claimed to be likely matched. Fifth, it determines the possible geometric transformations from the matched triangle pairs since triangles in an image undergo the same transformation as the image itself. The detailed steps are:

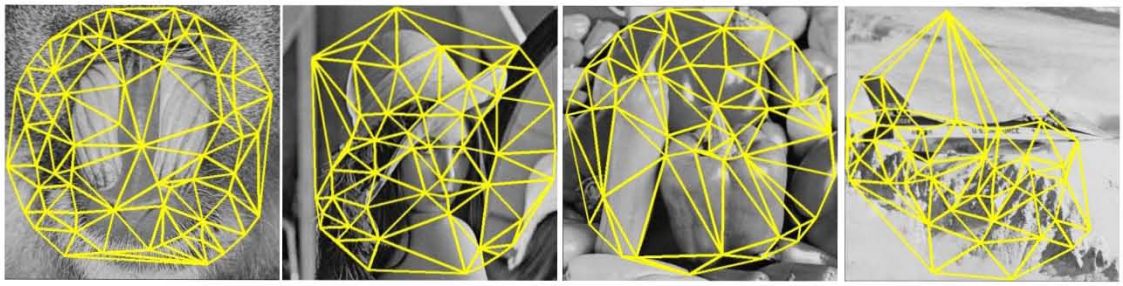
1. Calculate the scaling factor *SF* by resizing the probe triangle to the same size as the target matched triangle.
2. Calculate the translation factor *TF* by registering one of the vertices of the matched triangle pair.
3. Calculate the rotation factor *RF* by aligning the other two unregistered vertices 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. The estimated three-element-tuple (*SF*,

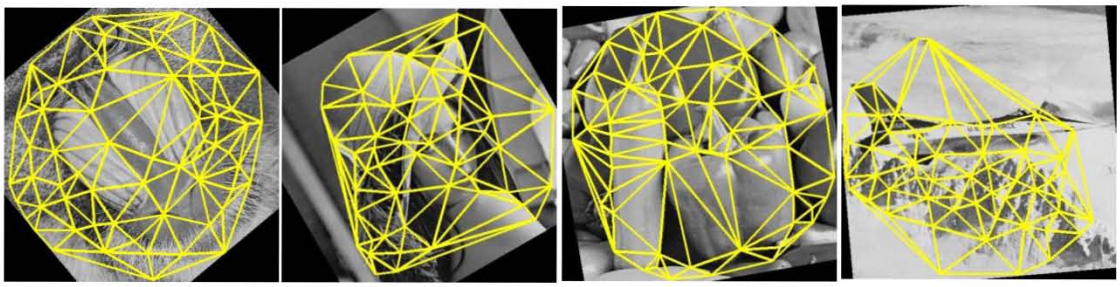
TF, *RF*) is then utilized to restore the probe image to be aligned with the original image.

Figure 3.7(a) and Figure 3.7(b) show the Delaunay tessellation results on robust Harris corner feature points of four original images and their probe images that underwent different rotation attacks, respectively. For example, a rotation of 40˚, 30˚, 10˚, and 5˚ is applied on the watermarked image of Baboon, Lena, Pepper, and Airplane, respectively. Figure 3.7(c) and Figure 3.7(d) demonstrate the matched triangle pairs as shown in the same color on the original and the probe images, respectively. The estimated transformation parameters for Baboon, Lena, Pepper, and Airplane, are (1, 1, 40˚), (1, 1, 30˚), (1, 1, 10˚), and (1, 1, 5˚), respectively. These angles are exactly the same as the ones used to distort the watermarked images and therefore can be used to restore the probe images to be aligned with the original image. The final restored images are shown in Figure 3.7(e). It clearly shows that the probe images undergoing different geometric attacks are correctly restored to align with their original images.

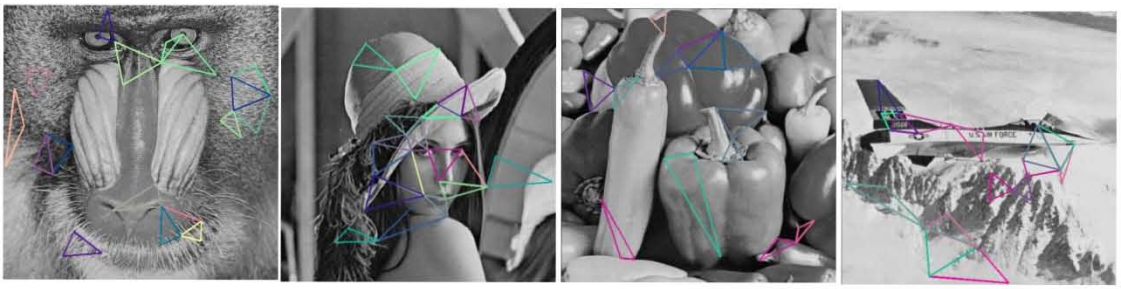
After restoring the probe image to its original position, the following DCT block-based watermark extraction steps are applied to extract the second watermark. 1) The aligned probe image is divided into 80×80 non-overlapping blocks. 2) Each block is divided into 8×8 non-overlapping sub-blocks. 3) Each sub-block is transformed into 8×8 DCT sub-block. 4) For each 80×80 non-overlapping block, every four sub-blocks are grouped together, and the watermark bit is extracted from each of these groups (embedding units) in the same order as generated in the embedding process. That is, each of four DC values in every embedding unit is modularly divided by the embedding strength α which is calculated by using the HVS method described in Section 3.1.2.3. The extraction function is:



(a)



(b)



(c)

(d)

(e)

Figure 3.7. Illustration of Delaunay tessellation results, matched Delaunay triangles, and final restoration results. (a) Delaunay tessellation results on the original images. (b) Delaunay tessellation results on the probe images. (c) Matched triangles on the original images. (d) Matched triangles on the probe images. (e) Final restored images to be aligned with the original images.

$$\tilde{w}_{k,i}' = \begin{cases} 1 & if\ DC_{k,i}' \bmod \alpha \geq \frac{1}{2}\alpha \\ -1 & if\ DC_{k,i}' \bmod \alpha < \frac{1}{2}\alpha \end{cases} \quad (15)$$
$$i = 1...4, k = 1...N$$

where, $DC_{k,i}'$ is the ith DC magnitude value in the kth embedding unit; $\tilde{w}_{k,i}'$ is one of the extracted bits in the embedding unit, and N is the length of the watermark. The final watermark bit $\hat{w}_k'$ of each embedding unit is decided by the majority value in the group $\tilde{w}_{k,j}'(i = 1...4)$.

Finally, it applies the watermark verification technique to decide the presence of the watermark. Specifically, the extracted watermark sequence is compared with the secret key generated embedded watermark sequence. A ratio of matched watermark bits and the total number of watermark bits is computed for each probe 80×80 block. We consider blocks with a ratio of larger than 0.84 (i.e., at most a 3-bit difference) containing a watermark. If at least two blocks contain a watermark, we claim that the presence of watermark in the probe image.

3.3 Watermark Detection Error

In our feature points-based watermarking scheme, we determine the two detection thresholds based on a fixed false-positive error probability. For an un-watermarked image, the extracted bits are treated as independent random variables with probability of 0.5. According to Bernoulli trials, the false-positive probability of an LCR is:

$$P_{fp\_LCR} = \sum_{i=T_{LCR}}^{L_w} (0.5)^{L_w} \times \left(\frac{L_w!}{i! \times (L_w - i)!} \right) \quad (16)$$

where T_{LCR} is the predefined threshold, i is the number of the matching bits, and L_w is

length of the first watermark sequence. In our system, T_{LCR} =17 and L_w=20. The false-positive probability of an image can then be expressed as follows:

$$P_{fp\_Image} = \sum_{j=T_{LCRn}}^{LCRN} \left(\frac{LCRN!}{j! \times (LCRN - j)!} \right) \times P_{fp\_LCR} \times \left(1 - P_{fp\_LCR}\right)^{LCRN-j} \qquad (17)$$

where T_{LCRn} is the predefined threshold, *LCRN* is the total number of LCRs in the probe image. In average, *LCRN* for the probe image is around 100. If we want the false-positive probability to be less than 10^{-4}, we need to set T_{LCRn} to be 3. In other words, if at least three LCRs can extract watermark of at most 3-bits difference from the embedded watermark, we claim the presence of a watermark in the probe image with the false-positive probability of 3.14×10^{-4}.

Similarly, the false-positive probability of an 80×80 block is:

$$P_{fp\_B} = \sum_{i=T_b}^{L_w} (0.5)^{L_w} \times \left(\frac{L_w!}{i! \times (L_w - i)!} \right) \qquad (18)$$

where T_b is the predefined threshold, i is the number of the matching bits, and L_w is length of the second watermark sequence. In our system, T_b =22 and L_w=25. The false-positive probability of an image can then be expressed as follows:

$$P_{fp\_Image} = \sum_{j=T_{bn}}^{LB} \left(\frac{LB!}{j! \times (LB - j)!} \right) \times P_{fp\_B} \times \left(1 - P_{fp\_B}\right)^{LB-j} \qquad (19)$$

where T_{bn} is the predefined threshold, *LB* is the total number of 80×80 blocks in the probe image. The value of *LB* for the probe image of size 512×512 is 36. If we want the false-positive probability to be less than 10^{-6}, we need to set T_{bn} to be 2. In other words, if at least two 80×80 blocks can extract a watermark of at most 3-bits difference from the embedded watermark, we claim the presence of a watermark in the probe image with the false-positive probability of 3.85×10^{-6}.

With these two predefined detection thresholds based on a fixed false-positive error probability, we can conclude the following:

- If only LCR-based watermarking detection procedure can find at least three LCRs contain a watermark (i.e., the extracted watermark contains at least 17 bits matched with the embedded watermark), the false positive probability of detecting non-watermarked images containing a watermark is 3.14×10^{-4}.
- If the block-based watermarking detection procedure can find at least two blocks contain a watermark (i.e., the extracted watermark contains at least 22 bits matched with the embedded watermark), the false positive probability of detecting non-watermarked images containing a watermark is 3.85×10^{-6}.
- If the LCR-based watermarking detection procedure can find at least three blocks containing a watermark and the block-based watermarking detection procedure can find at least two blocks contain a watermark, the false positive probability of detecting non-watermarked images containing a watermark is 1.21×10^{-9}.

CHAPTER 4

EXPERIMENTAL RESULTS AND COMPARISONS

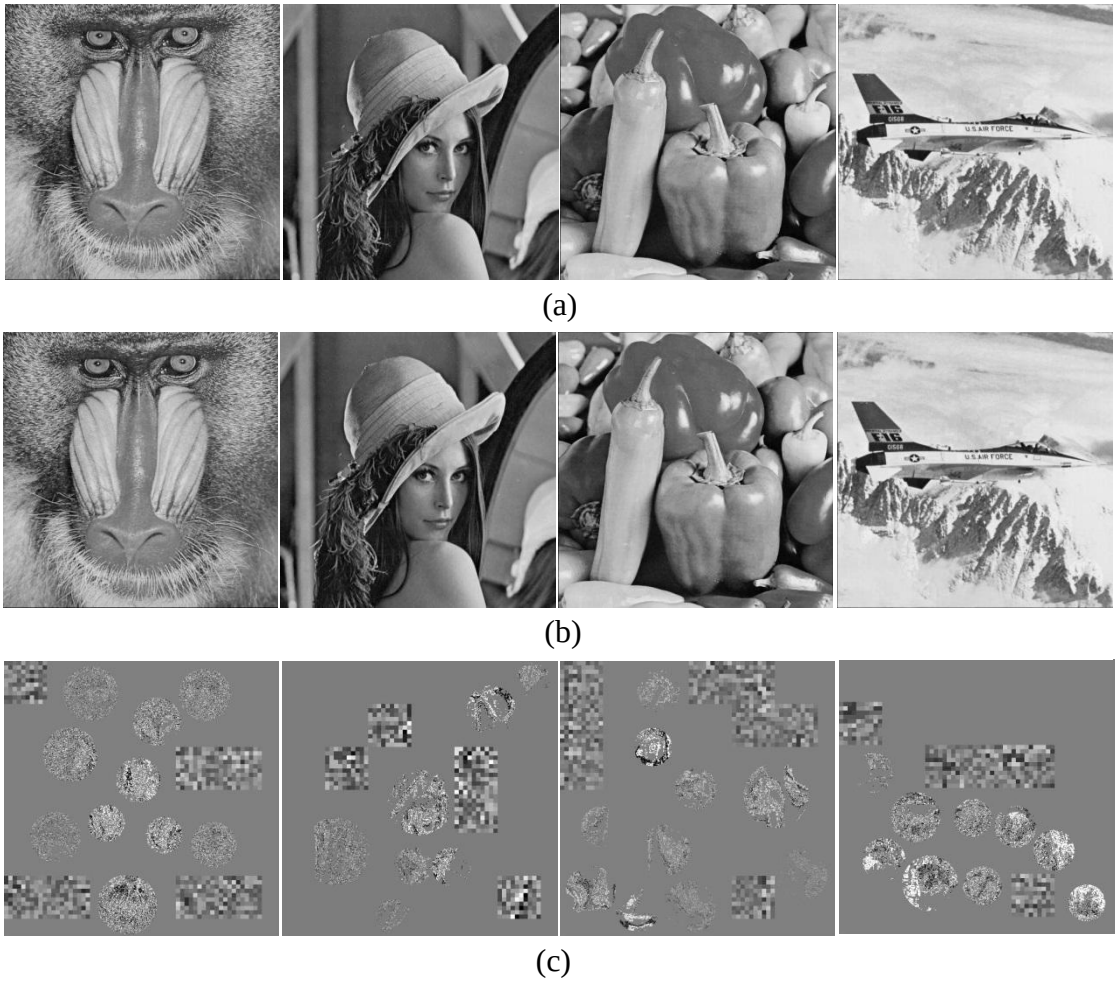
To evaluate the performance of the proposed watermarking scheme, we conduct a variety of experiments on various standard images using different kinds of attempted attacks. We first perform the watermark invisibility test using four 512×512 gray-scale images. These images are Baboon, Lena, Pepper, and Airplane. Although the goal of our watermarking scheme is to be RST-resilient, it is still working relative well under certain common image processing attacks. Therefore, we present not only the RST robustness of the proposed scheme but also its resistance to image processing attacks. In the simulation results section, we show our results under a variety of common image processing attacks and RST attacks. Intensive comparisons are finally performed with three well designed RST resilient watermarking schemes [16, 20, 22]. These three schemes use different methods to achieve the same goal – resistance to RST distortions.

4.1 Watermark Invisibility Test

We evaluate watermark invisibility on the following images: Baboon, Lena, Pepper, and Airplane. These four images correspond to several texture categories and have been extensively used in watermarking systems for benchmark comparison. For example, Baboon includes textured areas with high frequency components; Plane includes large homogeneous areas, whereas Lena has sharp edges and highly textured areas around the hair area; Pepper falls in a low-textured category. The PSNRs of these four watermarked images are 41.80, 46.62, 43.37, and 41.17, respectively. These PSNR values are all greater than 35.00db, which is the empirical value for the image without any perceivable degradation (i.e., a watermarked image as acceptable by human

perception) [28].

Figure 4.1 shows four original images, watermarked images, and scaled differences between watermarked and original images. One clearly sees that the watermarked image looks like the original image without any noticeable visual differences.



(a)

(b)

(c)

Figure 4.1. Illustration of the original images, watermarked images, and their differences. (a) Original images. (b) Watermarked images. (c) Scaled difference images between original images and the watermarked images.

4.2 Simulation Results

To evaluate the performance of the proposed watermarking scheme, we conducted experiments on different JPEG compression attacks and geometric attacks.

Figure 4.2 shows the detected LCRs and blocks for four images under no attack. We display all the LCRs detected to contain the watermark on purpose without applying the strategy summarized in Section 3.2.1 to remove the duplicated LCRs. We want to demonstrate the fact that most detected LCRs are non-overlapping to each other and the overlapped LCRs do have a sufficient overlapping that can be easily removed by the strategy summarized in Section 3.2.1. For this reason, we display all the detected LCRs in all the remaining figures. Figure 4.2 shows that our watermarking scheme successfully finds all embedding blocks and a majority of the embedding LCRs under no attacks.

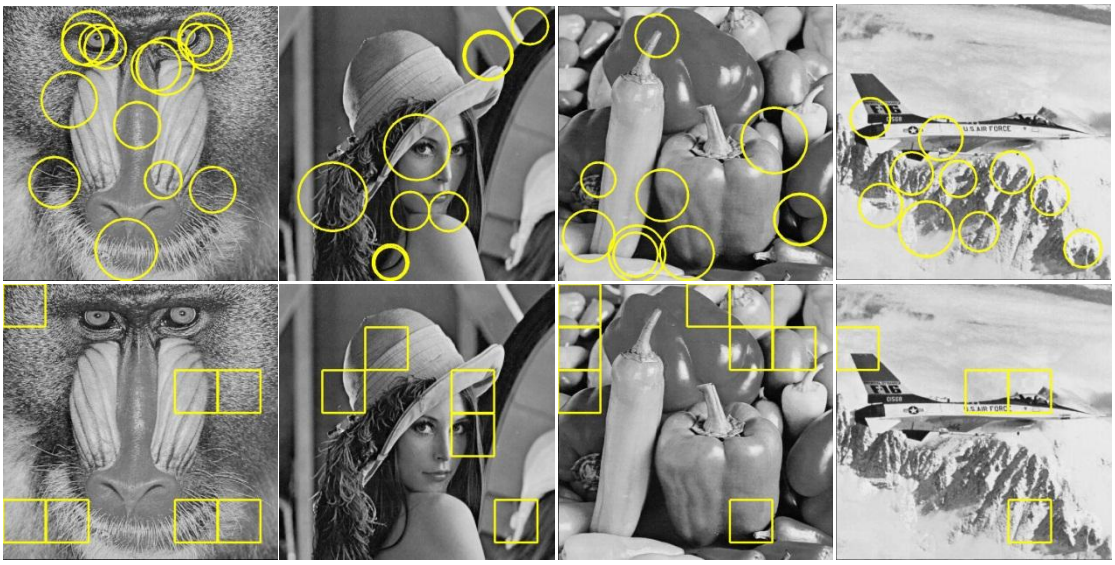


Figure 4.2. Watermark extraction results under no attack. LCR-based watermark extraction results (top row) and block-based watermark extraction results (bottom row).

5

Table 4.1 summarizes our watermarking detection results under various JPEG compression attacks. It clearly shows that our block-based watermarking scheme is resistant to JPEG compressions since a majority of the embedding blocks have been detected to contain the watermark down to JPEG compression quality factor of 40%. Our LCR-based watermarking scheme is resistant to JPEG compression quality factor of 80% or above. All the successful detection results are shown bolded based on the two predefined detection thresholds (i.e., at least 3 LCRs detected to contain watermark or at least 2 blocks detected to contain watermark). As a result, we claim that our proposed watermarking scheme is resistant to JPEG compression quality factor down to 40%.

Table 4.1. Ratios under JPEG Compression Attacks (LCR Ratio, Block Ratio).

Compression Quality Factor	Baboon	Lena	Pepper	Airplane
90%	(7/10, 7/7)	(5/7, 5/5)	(5/10, 8/8)	(10/9, 4/5)
80%	(5/10, 4/7)	(4/7, 5/5)	(4/10, 8/8)	(5/9, 4/5)
75%	(2/10, 7/7)	(3/7, 5/5)	(4/10, 8/8)	(3/9, 4/5)
70%	(2/10, 7/7)	(2/7, 5/5)	(1/10, 8/8)	(3/9, 4/5)
60%	(2/10, 2/7)	(1/7, 5/5)	(2/10, 8/8)	(0/9, 4/5)
50%	(0/10, 1/7)	(0/7, 5/5)	(1/10, 8/8)	(2/9, 4/5)
40%	(0/10, 7/7)	(1/7, 5/5)	(1/10, 8/8)	(0/9, 4/5)
30%	(1/10, 0/7)	(0/7, 5/5)	(1/10, 8/8)	(0/9, 1/5)

Figure 4.3 shows the detected LCRs and blocks for four images under two JPEG compression attacks: 75% JPEG compression and 80% JPEG compression. The figure clearly shows that a majority of the embedding blocks have been detected to contain the watermark. In other words, our block-based watermarking scheme provides more resistant to JPEG compressions.

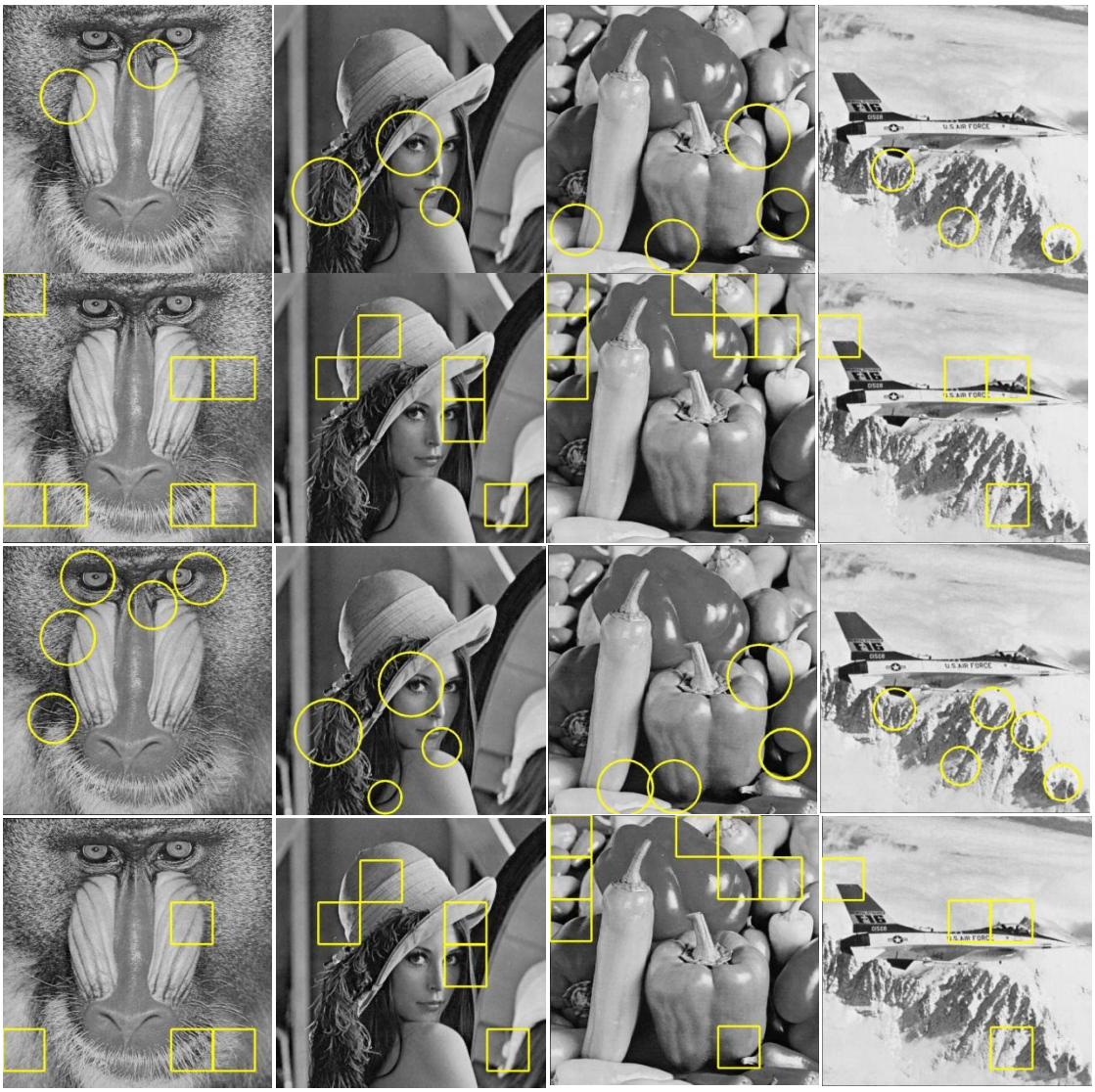


Figure 4.3. Watermark extraction results under JPEG compression. LCR-based and block-based watermark extraction results under 75% JPEG compression (top two rows), under 80% JPEG compression (bottom two rows).

Table 4.2 summarizes our watermarking detection results under various scaling attacks. One clearly sees that there is no clear winner between our block-based watermarking scheme and our LCR-based watermarking scheme. They complement each other well to achieve decent resistance to scaling attacks. All the successful detection results are shown bolded based on the two predefined detection thresholds. As a result, we claim that our proposed watermarking scheme is resistant to small scaling in

Table 4.2. Ratios under Scaling Attacks (LCR Ratio, Block Ratio).

Scaling Factor	Baboon	Lena	Pepper	Airplane
0.83	**(0/10, 3/7)**	**(1/7, 5/5)**	**(5/10, 0/8)**	**(0/9, 4/5)**
0.85	**(0/10, 2/7)**	**(3/7, 2/5)**	**(5/10, 4/8)**	(2/9, 1/5)
0.9	**(1/10, 7/7)**	(1/7, 0/5)	**(4/10, 0/8)**	**(1/9, 3/5)**
0.95	**(0/10, 4/7)**	**(2/7, 2/5)**	**(5/10, 4/8)**	**(1/9, 3/5)**
1.05	**(3/10, 1/7)**	**(5/7, 2/5)**	**(5/10, 6/8)**	**(5/9, 1/5)**
1.1	**(2/10, 2/7)**	**(3/7, 0/5)**	**(5/10, 5/8)**	**(3/9, 2/5)**
1.2	(2/10, 0/7)	(2/7, 0/5)	**(5/10, 0/8)**	**(5/9, 1/5)**
1.3	(2/10, 1/7)	(2/7, 0/5)	**(5/10, 6/8)**	**(5/9, 2/5)**
1.4	**(4/10, 0/7)**	(1/7, 0/5)	**(3/10, 8/8)**	**(3/9, 0/5)**
1.5	(2/10, 1/7)	**(4/7, 5/5)**	**(4/10, 8/8)**	**(3/9, 4/5)**
1.55	(0/10, 0/7)	(2/7, 0/5)	**(4/10, 0/8)**	**(5/9, 1/5)**
1.6	(0/10, 0/7)	(2/7, 0/5)	**(3/10, 2/8)**	(2/9, 1/5)
1.65	(0/10, 0/7)	(0/7, 0/5)	**(5/10, 8/8)**	**(4/9, 1/5)**
1.7	(0/10, 0/7)	**(3/7, 0/5)**	**(1/10, 7/8)**	**(5/9, 0/5)**
1.75	(0/10, 0/7)	(2/7, 0/5)	**(0/10, 8/8)**	**(3/9, 0/5)**
1.8	(0/10, 0/7)	(2/7, 0/5)	(0/10, 0/8)	**(5/9, 0/5)**

the range of 0.83 to 1.1. It works extremely well on the low-textured images such as Pepper and Airplane for a larger scale up to 1.75. However, it does not work on highly textured images such as Baboon and Lena.

Figure 4.4 shows the detected LCRs and blocks for four images under three scaling attacks including 0.95 scaling, 1.05 scaling, and 1.1 scaling. One clearly sees that LCR-based watermarking scheme and block-based watermarking scheme contribute equally to the watermark detection. In other words, our watermarking scheme provides more resistance to scaling attacks by combining the detection results from the LCR-based and block-based watermarking schemes.

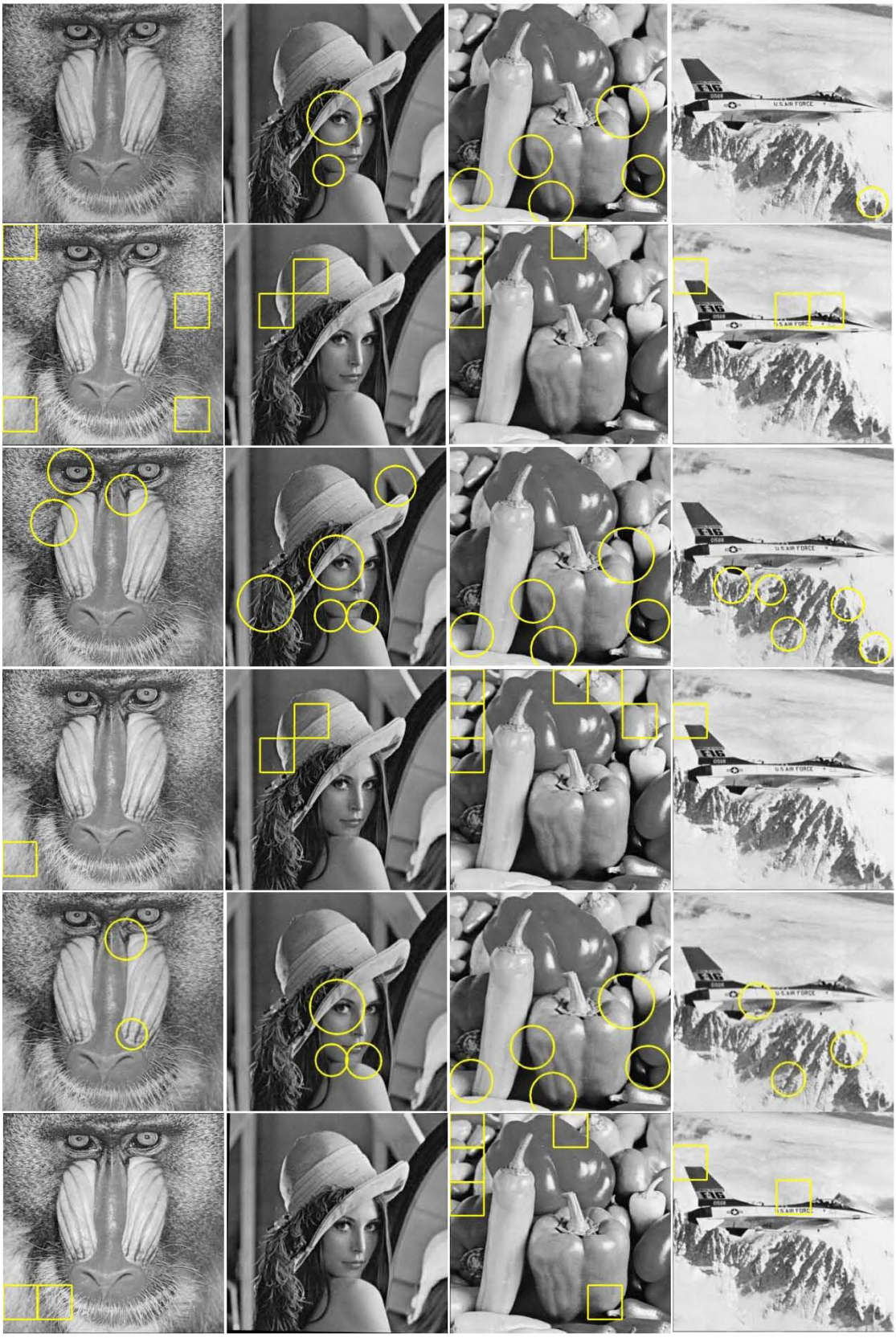


Figure 4.4. Watermark extraction results under scaling attacks. LCR-based and block-based watermark extraction results under 0.95 scaling attack (top two rows), 1.05 scaling attack (middle two rows), 1.1 scaling attack (bottom two rows).

Table 4.3 summarizes our watermarking detection results under various rotation attacks. One clearly sees that our LCR-based watermarking scheme works well under all rotation attacks, and our block-based watermarking scheme works well under all rotation attacks except six cases marked as italic and bold. Based on the two predefined detection thresholds, we claim that our proposed watermarking scheme is resistant to all rotation attacks.

Figure 4.5 shows the detected LCRs and blocks for four images under two rotation attacks including 1˚ (a small rotation angle) and 15˚ (a relatively large rotation angle) rotations. One clearly sees that LCR-based watermarking scheme detects a majority of embedded LCRs as containing a watermark while block-based watermarking schemes may not find any embedded blocks.

Table 4.4 summarizes our watermarking detection results under various translation attacks. One clearly sees that our LCR-based and block-based watermarking schemes work well under all translation attacks. Based on the two predefined detection thresholds, we claim that our proposed watermarking scheme is resistant to all translation attacks.

Table 4.3. Ratios under Rotation Attacks (LCR Ratio, Block Ratio)

Rotation Angle	Baboon	Lena	Pepper	Airplane
1˚	(9/10, 4/7)	***(5/7, 0/5)***	(9/10, 2/8)	(10/9, 2/5)
2˚	(7/10, 4/7)	(5/7, 3/5)	(8/10, 3/8)	(8/9, 3/5)
3˚	(9/10, 4/7)	***(6/7, 1/5)***	(6/10, 2/8)	(10/9, 2/5)
5˚	(9/10, 4/7)	(5/7, 5/5)	***(8/10, 1/8)***	(9/9, 2/5)
10˚	(8/10, 4/7)	(4/7, 4/5)	(7/10, 2/8)	(10/9, 2/5)
15˚	***(9/10, 0/7)***	(5/7, 2/5)	(3/10, 4/8)	(9/9, 3/5)
30˚	(9/10, 4/7)	(4/7, 3/5)	(4/10, 3/8)	(9/9, 2/5)
45˚	(9/10, 4/7)	(4/7, 3/5)	(5/10, 2/8)	***(9/9, 0/5)***
60˚	(9/10, 4/7)	(5/7, 4/5)	(6/10, 4/8)	(9/9, 2/5)
75˚	(10/10, 4/7)	(4/7, 2/5)	(6/10, 4/8)	(8/9, 2/5)
90˚	(10/10, 4/7)	***(5/7, 0/5)***	(7/10, 8/8)	(10/9, 4/5)

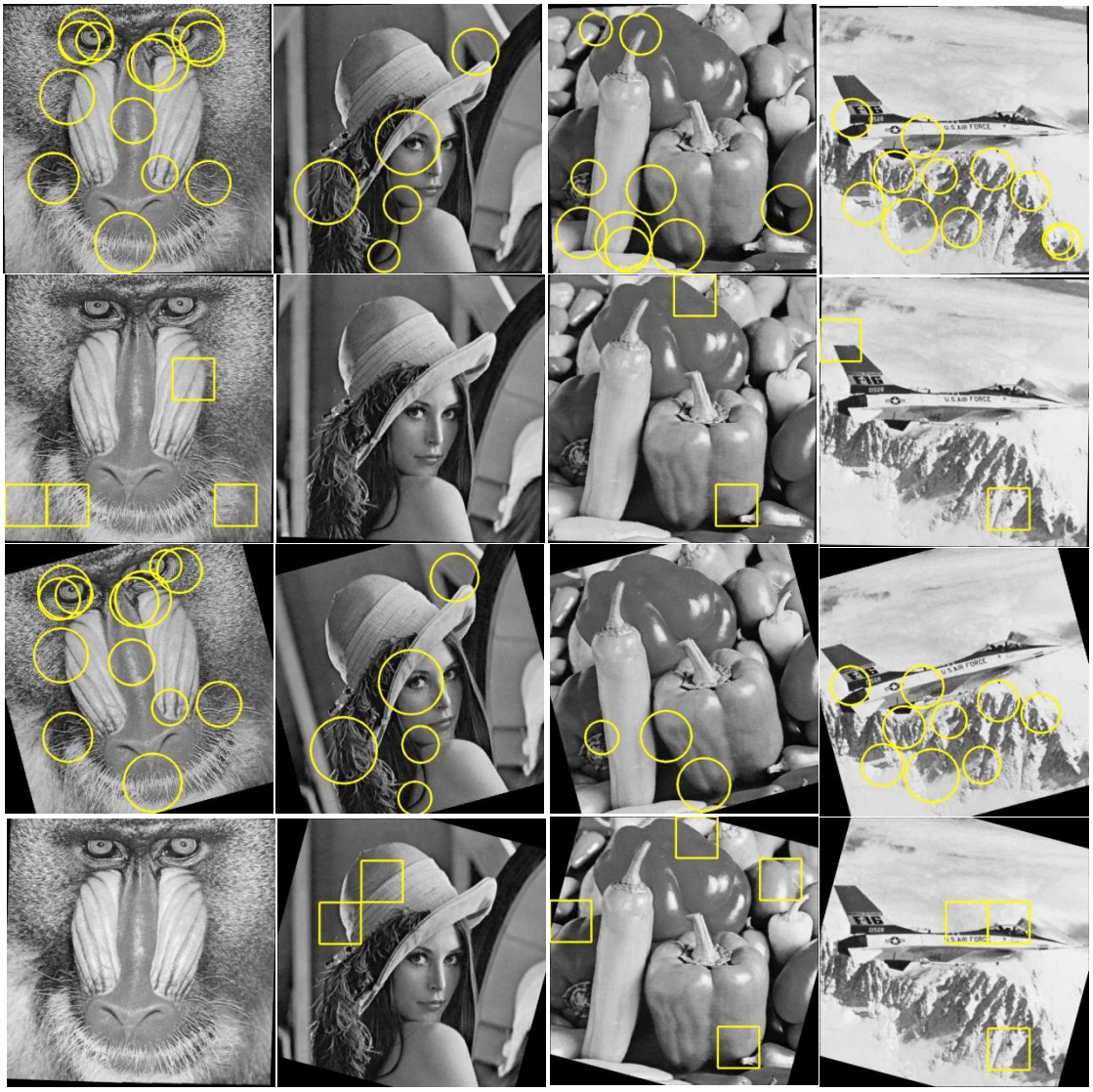


Figure 4.5. Watermark extraction results under rotation attacks. LCR-based and block-based watermark extraction results under 1˚ rotation attack (top two rows) and 15˚ rotation attack (bottom two rows).

Table 4.4. Ratios under Translation Attacks (LCR Ratio, Block Ratio).

Translation	Baboon	Lena	Pepper	Airplane
Shift 25 rows	(9/10, 6/7)	(5/7, 5/5)	(7/10, 5/8)	(10/9, 4/5)
Shift 40 rows	(9/9, 6/7)	(6/7, 5/5)	(7/10, 5/8)	(10/9, 4/5)
Shift 80 rows	(6/10, 9/7)	(4/7, 5/5)	(7/10, 5/8)	(10/10,4/5)

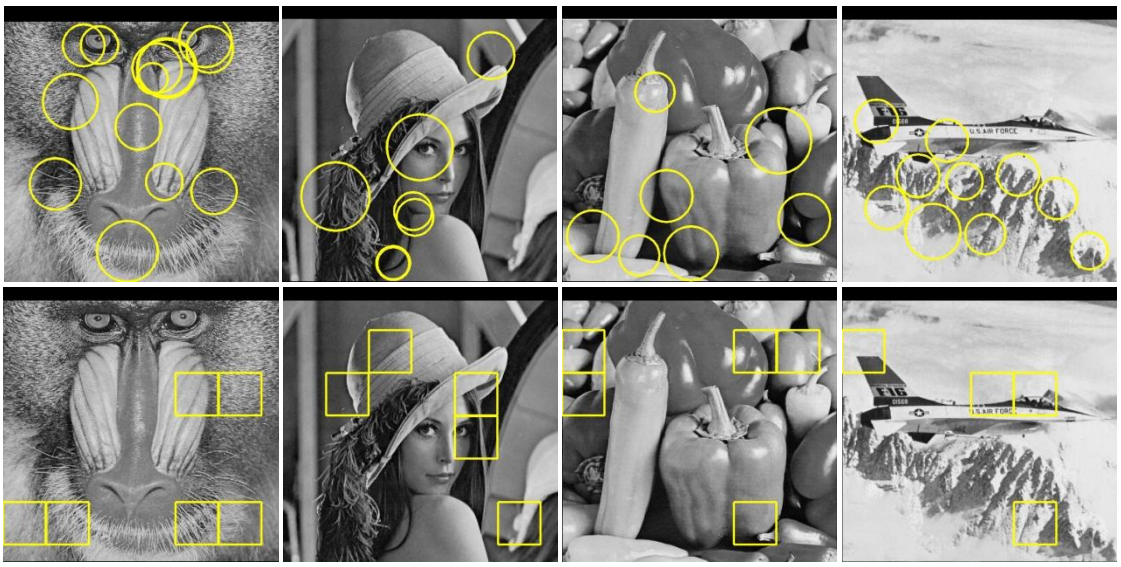


Figure 4.6. Watermark extraction results under 25 rows translation attack. LCR-based watermark extraction results (top row) and block-based watermark extraction results (bottom row)

Figure 4.6 shows the detected LCRs and blocks for four images under translation attack. One clearly sees that both LCR-based and block-based watermarking schemes detect a major of embedded LCRs and blocks as containing watermark.

Table 4.5 summarizes our watermarking detection results under various combined RST attacks. One clearly sees that our LCR-based watermarking scheme complements with our block-based watermarking scheme to achieve robustness against all attacks except one case shown in italic and bold. Our extensive experiments show our scheme is resilient against the combined RST attacks for small scaling and the JPEG compression quality factor down to an 80% quality factor.

Figure 4.7 shows the detected LCRs and blocks for four images under a combined attack. One clearly sees that the LCR-based and block-based watermarking schemes complement each other to achieve robustness against the combined RST attacks.

Table 4.5. Ratios under Combined RST Attacks (LCR Ratio, Block Ratio).

	Baboon	Lena	Pepper	Airplane
No attack	(9/10, 7/7)	(7/7,5/5)	(8/10, 8/8)	(9/9, 5/5)
(1˚, 1.0, 90%)	(8/10, 4/7)	(4/7, 0/5)	(4/10, 3/8)	(8/9, 2/5)
(1˚, 1.0, 80%)	(1/10, 4/7)	(2/7, 4/5)	(5/10, 2/8)	(4/9, 2/5)
(5˚, 1.0, 90%)	(7/10, 5/7)	(3/7, 5/5)	(6/10, 1/8)	(7/9, 1/5)
(5˚, 1.0, 80%)	(0/10, 4/7)	(1/7, 5/5)	(3/10, 3/8)	***(2/9, 1/5)***
(10˚, 1.0, 90%)	(6/10, 4/7)	(4/7, 4/5)	(6/10, 3/8)	(8/9, 1/5)
(15˚, 1.0, 90%)	(8/10, 3/7)	(3/7, 2/5)	(3/10, 4/8)	(8/9, 1/5)
(15˚, 1.1, 100%)	(3/10, 0/7)	(2/7, 2/5)	(3/10, 0/8)	(3/9, 0/5)
(30˚, 1.0, 90%)	(5/10, 0/7)	(2/7, 3/5)	(2/10, 5/8)	(8/9, 1/5)
(30˚, 1.1, 100%)	(3/10, 0/7)	(3/7, 3/5)	(3/10, 1/8)	(5/9, 0/5)
Row, Column removal (5,17)	(8/10, 2/7)	(7/7, 5/5)	(9/10, 8/8)	(10/9, 4/4)
Row, Column Removal(5,17) + JPEG70%	(2/10, 7/7)	(2/7, 5/5)	(1/10, 8/8)	(3/9, 4/5)

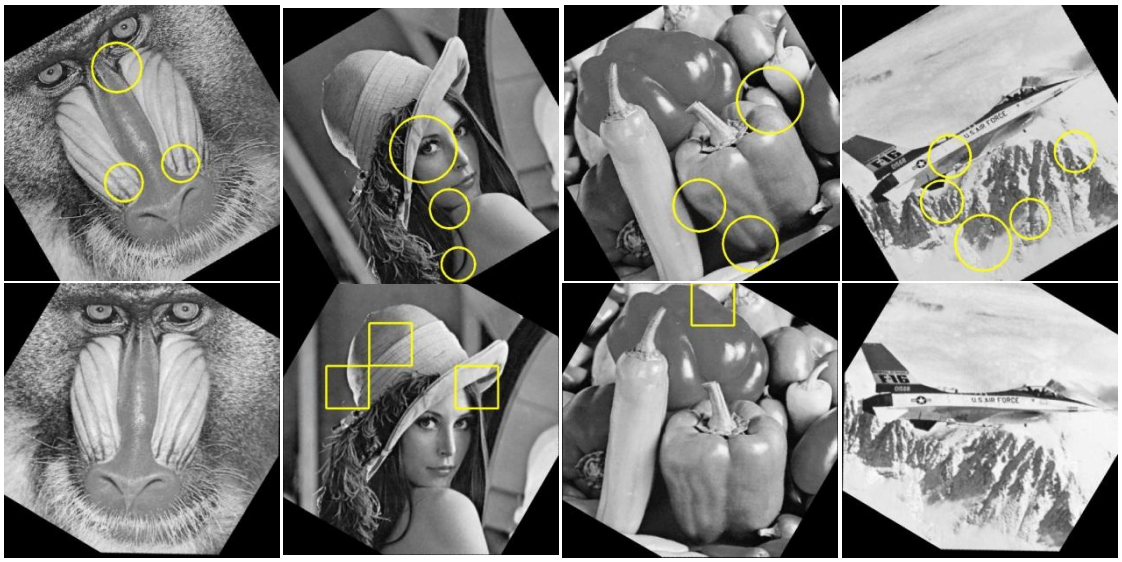


Figure 4.7. Watermark extraction results under combined attack (100% JPEG compression, 1.1 scaling, and 30˚ rotation). LCR-based watermark extraction results (top row) and block-based watermark extraction results (bottom row)

4.3 Comparison with Other Methods in the Literature

The results of the proposed method are compared with Deng's method [16] (histogram-based), Tang's method [20], and Bas's method [22]. These methods are chosen because all of them belong to the feature-based watermarking group, and Deng's method is histogram-based method as well.

Table 4.6 compares our system with Deng's method [16] using the same experiments summarized in [16]. The table shows that our results are comparable with Deng's results. However, our system does not work well under a low JPEG compression quality factor such as 50% or 30%, nor does it work well under a large scaling attack. The four unsuccessful detections are shown in italic and bold. One advantage of our system is that it allows two kinds of watermark embedding in two different regions (i.e., LCRs and blocks). That is, the payload is higher than the payload of Deng's method. Another advantage of our system is that our system is more efficient than Deng's method since Deng's detection step searches for a 2×2 neighborhood of each Harris-Laplace feature point to find the best match. This neighborhood search is time consuming since there are lots of Harris-Laplace feature points.

Table 4.7 compares our system with Tang's method [20] using the same experiments summarized in [20]. The table shows that our method fails to detect watermark under four kinds of attacks, namely, a JPEG compression of quality factor of 50%, a JPEG compression of quality factor of 30%, a rotation 5˚ plus cropping and scaling, and removing 5 rows and 17 columns plus a JPEG compression of quality factor of 70%.

Table 4.6. Comparison of Our Method in Terms of LCR Ratio and Block Ratio with Deng's Method [16].

Attacks	Baboon		Lena		Pepper		Plane	
	Ours	[16]	Ours	[16]	Ours	[16]	Ours	[16]
Crop 10% off	8/10 5/7	10/17	4/7 3/5	7/13	6/10 7/8	7/18	5/9 4/5	5/14
Scaling 1.5	***2/10*** ***1/7***	11/17	4/7 5/5	10/13	4/10 8/8	13/18	3/9 4/5	8/14
Rotation 5˚	9/10 0/7	8/17	5/7 5/5	5/13	8/10 1/8	10/18	9/9 2/5	6/14
Rotation 30˚	9/10 0/7	8/17	4/7 3/5	5/13	4/10 3/8	7/18	9/9 2/5	6/14
Row, Column removal (5,17)	8/10 2/7	7/17	7/7 5/5	7/13	9/10 8/8	8/18	10/9 4/4	6/14
Gaussian fitter 3×3	2/10 7/7	8/17	7/7 5/5	5/13	5/10 8/8	11/18	6/9 4/5	6/14
Median filter 3×3	6/10 3/7	12/17	7/7 5/5	7/13	7/10 8/8	16/18	10/9 4/5	10/14
Rotation5˚+Cropping+JPEG70%	0/10 6/7	6/17	1/7 4/5	7/13	3/10 3/8	8/18	0/9 3/5	6/18
Row, Column Removal (5,17) + JPEG70%	2/10 7/7	7/17	2/7 5/5	7/13	1/10 8/8	8/18	3/9 4/5	6/14
JPEG 90%	7/10 7/7	16/17	5/7 5/5	12/13	5/10 8/8	17/18	10/9 4/5	13/14
JPEG 70%	2/10 7/7	15/17	2/7 5/5	12/13	1/10 8/8	17/18	3/9 4/5	12/14
JPEG 50%	***0/10*** ***1/7***	15/17	0/7 5/5	11/13	1/10 8/8	15/18	2/9 4/5	12/14
JPEG 30%	***1/10*** ***0/7***	14/17	0/7 5/5	10/13	1/10 8/8	16/18	***0/9*** ***1/5***	11/14

Tang' method fails to detect watermarks under four kinds of attacks, namely, rotation of 1˚ plus cropping and scaling, rotation of 5˚ plus cropping and scaling, a JPEG compression of a quality factor of 30%, and removing 5 rows and 17 columns. However, our ratios are generally larger than Tang's, which indicates our system is more likely to extract feature points from the embedded regions. In addition, Tang's method can only

resist small angle rotations, while our method can resist large rotation angles as summarized in Table 4.3.

Table 4.7 Comparison of Our Method (LCR Ratio, Block Ratio) with Tang's Method [20].

Attacks	Lena		Baboon		Pepper	
	Ours	[20]	Ours	[20]	Ours	[20]
Watermarked Image	(7/7, 5/5)	7/8	(9/10,7/7	10/11	(8/10,8/8)	4/4
Median filter 2×2	(3/7, 2/5)	1/8	(1/10,2/7)	6/11	(5/10,2/8)	1/4
Median filter 3×3	(7/7, 5/5)	1/8	(6/10,3/7)	2/11	(7/10 ,8/8)	1/4
Gaussian filter	(7/7, 5/5)	5/8	(2/10,7/7)	8/11	(5/10,8/8)	1/4
Additive uniform noise (Scale=0.1)	(7/7,5/5)	5/8	(8/10,7/7)	6/11	(8/10,8/8)	4/4
Additive uniform noise (Scale=0.15)	(7/7,5/5)	4/8	(8/10,7/7)	4/11	(7/10,8/8)	2/4
Additive uniform noise (Scale=0.2)	(6/7.5/5)	1/8	(8/10,7/7)	5/11	(8/10,8/8)	1/4
JPEG 80%	(4/7,5/5)	6/8	(5/10,4/7)	9/11	(4/10,8/8)	3/4
JPEG 70%	(2/7,5/5)	7/8	(2/10,7/7)	11/11	(1/10,8/8)	3/4
JPEG 60%	(1/7,5/5)	6/8	(2/10,2/7)	7/11	(2/10,8/8)	1/4
JPEG 50%	(0/7,5/5)	5/8	***(0/10,1/7)***	7/11	(1/10,8/8)	3/4
JPEG 40%	(1/7,5/5)	3/8	(0/10,7/7)	5/11	(1/10,8/8)	1/4
JPEG 30%	(0/7,5/5)	2/8	***(1/10,0/7)***	4/11	(1/10,8/8)	***0/4***
Gaussian filter 3x3+JPEG 90%	(1/7,5/5)	5/8	(0/10,7/7)	8/11	(2/10,8/8)	2/4
Rotation 1˚ +Cropping	(5/7,0/5)	3/8	(9/10,4/7)	3/11	(9/10,2/8)	2/4
Rotation 1˚ +Cropping+Scale	(4/7,0/5)	***0/8***	(8/10,4/7)	4/11	(4/10, 3/8)	2/4
Rotation 5˚ +Cropping+Scale	(3/7, 2/5)	***0/8***	***(1/10,0/7)***	***0/11***	(5/10, 5/8)	***0/4***
Removal (1,5)	(7/7,5/5)	3/8	(9/10,7/7)	6/11	(8/10,8/8)	3/4
Remove 1 row, 5 columns+JPEG70%	(2/7,5/5)	4/8	(2/10,7/7)	6/11	(1/10,8/8)	3/4
Remove 5 rows, 17 columns	(5/7,0/5)	***0/8***	(8/10,4/7)	3/11	(9/10,8/8)	1/4
Remove 5 rows, 17 columns+JPEG70%	***(1/7,0/5)***	1/8	(0/10,4/7)	3/11	(3/10, 1/8)	1/4

Table 4.8 we further compared our system with Bas's system [22]. Our results show that our system achieves better scaling resistance than the Bas's system. However, Bas's system performs better under the JPEG compression attacks of a quality factor of 50%.

Table 4.8 Comparison of Our Method with Bas's Method [22].

Attacks	Plane		Baboon		Lena		Pepper	
	Ours	[22]	Ours	[22]	Ours	[22]	Ours	[22]
Rot 10	OK	OK	OK	OK	OK	OK	OK	OK
Scale 80	OK	DEFEAT	OK	OK	OK	OK	OK	OK
JPEG 50	OK	OK	DEFEAT	OK	OK	OK	OK	OK

CHAPTER 5

CONCLUSIONS

In this project, we propose a novel and robust geometric distortion resilient digital watermarking approach. The major contributions consist of:

- Applying several pre-attacks to select salient and robust SIFT feature points.
- Applying a histogram bin quality-based strategy to quickly find the best non-overlapping LCRs that contain a sufficient number of pixels, for embedding watermarks.
- Applying a histogram relationship-based embedding strategy to embed one watermark using the histogram and the mean statistically independent of the pixel positions.
- Applying a DCT-based visual model to embed the other watermark in highly textured blocks determined by the robust Harris corner detector.
- Applying Delaunay tessellation and Delaunay triangle matching to restore the probe image to be aligned with the original image to make the watermarking system more resilient to geometric attacks and JPEG compression attacks.

The proposed method is robust against a wide variety of tests as indicated in the experimental results. In particular, it is more robust against rotation attacks and translation attacks than other feature-based watermarking techniques. It works relatively well under scaling attacks except for images with high textures, such as the Baboon image used in the experiments. It works well only under a JPEG compression quality factor down to 60%. Our extensive experiments also show that our system achieves comparable performance to the peer systems. Our approach can be further improved by

developing a more reliable feature extraction method and a more stable embedding function for LCR-based histogram relationship-based embedding and block-based DC component embedding methods under combined geometric distortions.

REFERENCES

[1] Voyatzis, G. and Pitas, I. Protecting digital image copyrights: A framework. *IEEE Computer Graphics and Applications 19*, 1 (1999), 18 – 24.

[2] Mannos, J.L. and Sakrison, D.J. The effect of a visual fidelity criterion in the encoding of images. *IEEE Trans. Information Theory 20* (Jul. 1974), 525-536.

[3] Cox, I.J., Miller, M.L., and Bloom, J.A. *Digital Watermarking and Steganopraphy*, 2nd ed. , Elsevier.

[4] Peticolas, F., Anderson, R., and Kuhn, M. Attacks on copyright marking systems. In *Proceedings 2nd Workshop Information Hiding*, 1998, 218-238.

[5] Kutter, M., Bhattacharjee, S.K., and T. Ebrahimi, T. Towards second generation watermarking scheme. In *Proceedings IEEE Conference on Image Processing*, 1999, 320–323.

[6] Alghoniemy, M. and Tewfik, A. Geometric distortion correction through image normalization. In *Proceedings of the IEEE Conference on Multimedia and Expo*, vol. 3, 2000, 1291–1294.

[7] Dong, P. and Galatsanos, N. Affine transformation resistant watermarking based on image normalization. In *Proceedings of the IEEE Conference on Image Processing*, 2002, 489–492.

[8] Kim, H. and Lee, H-K. Invariant image watermark using Zernike moments. *IEEE Trans. Circuits, Systems and Video Technology 13*, 8 (2003), 766–775.

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

[10] Xin, Y.Q., Liao, S., and Pawlak, M. Geometrically robust image watermark via pseudo-Zernike moments. In *Proceeding of the IEEE Canadian Conference on Electrical and Computer Engineering 2* (2004), 939–942.

[11] Zhang, L., Qian, G., Xiao, W., and Li, Z. Geometric invariant blind image watermarking by invariant Tchebichef moments. *Optics Express 15*, 5 (2007), 2251–2261.

[12] Xiang, S., Kim, H.J., and Huang, J. Invariant image watermarking based on statistical features in the low-frequency domain. *IEEE Trans. Circuits and Systems for Video Technology 18*, 6 (2008), 777–790.

[13] DColtuc, D. and Bolon, P. Robust watermarking by histogram specification. In *Proceedings of the IEEE Conference on Image Processing 2* (1999), 236–239.

[14] Chareyron, G., Macq, B., and Tremeau, A. Watermarking of color images based on segmentation of the XYZ color space. In *Proceeding of the 2nd European Conference on Color in Graphics, Imaging and Vision*, 2004, 178–182.

[15] Lin, C-H., Chan, D-Y., Su, H., and Hsieh, W-S. Histogram-oriented watermarking algorithm: Colour image watermarking scheme robust against geometric attacks and signal processing. *IEEE Proceedings Vision, Image Signal Processing 53*, 4 (2006), 483–492.

[16] Deng, C., Gao, X., Li, X., and Tao, D. Local histogram based geometric invariant image watermarking. *Signal Processing 90*, 12 (2010), 3256-3264.

[17] Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Journal Computer Vision 60*, 2 (2004), 91–110.

[18] Li, L., Qian, J., and Pan, J.S. Characteristic region based watermark embedding with RST invariance and high capacity. *Journal of Electronics and Communications 65* (2011), 435-442.

[19] Seo, S. and Yoo, C.D. Localized image watermarking based on feature points of scale-space representation. *Pattern Recognition 37*, 7 (2004), 1365–1375.

[20] Tang, C.W. and Hang, H.M. A feature-based robust digital image watermarking scheme. *IEEE Trans. Signal Processing 51*, 4 (Apr. 2003), 950-959.

[21] Alghoniemy, M. and H. Tewfik, A.H. Geometric distortion correction through image normalization. In *Proceedings IEEE International Conference on Multimedia and Expo 3* (2000), 1291–1294.

[22] Bas, P., Chassery, J.M., and Macq, B. Geometrically invariant watermarking using feature points. *IEEE Trans. on Image Processing 11*, 9 (Sept. 2002), 1014-1028.

[23] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. Speeded-up robust features. *Computer Vision and Image Understanding* 110, 3 (2008), 346--359.

[24] Qi, X. and Qi, J. A robust content-based digital image watermarking scheme. *Signal Processing 87*, 6 (2007) 1264-1280.

[25] Harris, C. and Stephen, M. A combined corner and edge detector. In *Proceedings 4th Alvey Vision Conference*, 1988, 147-151.

[26] M. Eyadat, Factors that affect the performance of the DCT-block based image watermarking algorithms. In *Proceedings of International Conference on Information Technology: Coding and Computing 1* (2004), 650-654.

[27] Bertin, E., Marchand-Maillet, S., and Chassery, J.M. *Optimization in Voronoi Diagrams*. Kluwer, 1994.

[28] Hsieh, M.S. and Tseng, D.C. Perceptual digital watermarking for image authentication in electronic commerce. *Electronic Commerce Research*, 4 (2004), 157-170.