Geometrically Invariant Watermarking Scheme Based on Local Feature Points

Li Jing
School of Computer and Information Engineering, Henan University of Economics and Law, Zhengzhou, China
Email: jingli776@126.com

Xiaowen Zhang
Department of Computer Science, University College Cork, College Road, Cork, Ireland
Email: skyof45@yahoo.com.cn

Abstract—Based on local invariant feature points and cross ratio principle, this paper presents a feature-point-based image watermarking scheme. It is robust to geometric attacks and some signal processes. It extracts local invariant feature points from the image using the improved scale invariant feature transform algorithm. Utilizing these points as vertexes it constructs some quadrilaterals to be as local feature regions. Watermark is inserted these local feature regions repeatedly. In order to get stable local regions it adjusts the number and distribution of extracted feature points. In every chosen local feature region it decides locations to embed watermark bits based on the cross ratio of four collinear points, the cross ratio is invariant to projective transformation. Watermark bits are embedded by quantization modulation, in which the quantization step value is computed with the given PSNR. Experimental results show that the proposed method can strongly fight more geometrical attacks and the compound attacks of geometrical ones.

Index Terms—digital watermark, invariant feature points, the cross ratio of four collinear points, geometrical attacks, quantization modulation

I. INTRODUCTION

Geometric attacks means watermarked image is transformed by some geometrical distortions, such as rotation, scaling, translation (RST), random bend attack (RBA), and shearing etc. Geometric attacks induce synchronization errors between the original and the extracted watermark during the detection process, and the inserted watermark can not be detected correctly although it is still present in watermarked image. Up to now a few algorithms have presented the topic of how to achieve robustness against geometric attacks, which can be approximately classified into the following categories: watermarking based on invariant transform[1,2], watermarking based on synchronization templates[3,4], watermarking based on invariant moments[5,6], and watermarking based on feature points[7,8,9,10]. Compared with other kinds of algorithms, watermarking based on feature points has better performance against geometric attacks. It is not only robust to RST attack and local geometric attacks (for example RBA and cropping), but also to combination of more geometric attacks. For this reason watermarking based on feature points has been a researching focus in recent years.

Feature-point-based watermarking uses local invariant feature points to decided local feature regions, watermark is inserted these local regions repeatedly. The watermark inserted positions combined with these invariant feature points, in this way keeping watermarking synchronization. The stability of feature points to geometric transform and signal processes decides the robustness of watermarking system. Bas et al. [7] used Harris corner detecting method to extract feature points from image. Tang and Hang [8] extract feature points by Mexican hat wavelet scale interaction. Wang et al. [9] use Harris-Laplace method and Lee et al.[10] use Scale Invariant Feature Transform (SIFT) method. The stability of extracted feature points by these methods is very different. Harris corner points are robust to rotation, but not to scaling and affine transform. Feature points extracted by Mexican hat wavelet scale interaction are robust to signal processes, but not to geometric transform. Compared to these methods, feature points from Harris-Laplace and SIFT have better robustness [11].

Other important factors of Feature-point-based watermarking are the selection of local feature regions and the watermark inserting and extracting method. Many algorithms [9, 10] choose circle patches centering local feature points to insert watermark. They are robust to rotation and scaling, but not to shearing of affine transform. When the watermarked image is transformed by shearing, part of inserted watermark bits will beyond the detected circle patches and watermark can not be extracted correctly. Tang and Hang [8] used normalized circle patches to resist affine transform. However, the normalization is sensitive to the image contents used, so the robustness of these patches will decrease when the image is distorted. Up to now these algorithms only
This paper utilizes improved SIFT algorithm and invariance of cross-ratio in projective transform, proposed a watermarking scheme. The proposed scheme belongs to feature-point-based watermarking, is robust to geometric attack. It extracts stable feature points using the improved SIFT algorithm, and optimizes the distribution of feature points. Then choose qualified quadrilateral regions to insert watermark. Every region is decided by four feature points. The watermark is inserted into every region repeatedly. The location to insert watermark bit is based on invariance of cross-ratio, and using quantization modulation method to insert. Experimental results show the proposed scheme is not only robust to general signal processes, but also to many geometrical distortions, such as affine transform, projective transform, cropping and their combination.

The rest of this paper is organized as follows. Section 2 describes the feature points extracting and adjusting method used in the proposed scheme. Section 3 reviews cross-ratio theory of four collinear points and given two lemmas, which is the theory basis of watermark insert method. In Section 4, we present watermark inserting process. Section 5 covers the details of the watermark detection procedure. Simulation results in Section 6 will show the performance of our scheme. Finally, Section 7 concludes this presentation.

II. EXTRACTING FEATURE POINTS AND OPTIMIZING THEIR DISTRIBUTION

This paper utilizes SIFT algorithm to extract feature points. SIFT feature is extracted by considering local image properties and is invariant to rotation, scaling, translation, and partial illumination changing. And also robust to a substantial range of affine distortion, change in 3D viewpoint, addition of noise[12].

The basic idea of the SIFT is to extract features through a staged filtering that identifies stable points in the scale space. It first selects candidates for features by searching for peaks in the scale space of the difference-of-Gaussians (DoG) function, then localizes each feature using measures of its stability. The candidate locations that have a low contrast or are poorly localized along edges are removed by measuring the stability of each feature using a 2×2 Hessian matrix $H$ as follows:

$$
\text{stability} = \frac{\text{Tr}(H)^2}{\text{Det}(H)} < \left( \frac{r + 1}{r} \right)^2
$$

$$
\text{Where } H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}
$$

(1)

Here $r$ is the ratio of the largest to the smallest eigenvalue and is used to control stability. They use $r=10$. The quantities $D_{xx}$, $D_{xy}$, and $D_{yy}$ are the derivatives of the scale space images.

The next work is to assign orientations based on local image gradient directions, at last give a descriptor for every SIFT feature point. Our scheme only use SIFT points to decide quadrilateral regions when inserting and detecting watermark. So we abnegate constructing feature vector process and save computing time.

SIFT algorithm generates large numbers of features points that densely cover the image. A typical image of size 500×500 pixels will give rise to about 2000 stable features points, and these feature points densely covering structure complex regions of image. And the feature points used in watermark inserting and extracting should be stable and be distributed equally. So we improved SIFT algorithm to get stable feature points with optimized distribution, removing those feature points that are susceptible to watermark attacks.

Combined our work, we change the procedure of SIFT algorithm, and the following is our improving measures.

- Using a Gaussian filter to blur the original image before feature points extraction. By this preprocess, it can reduce the interference of noise and increase the robustness of extracted feature points.
- In SIFT algorithm we leave out the step of representing feature descriptor that is 128 element feature vector for each feature. Only remain feature points detector, which reduces largely calculating time without destroying our scheme.
- In order to improve the robustness of feature points, properly increase the sampling frequency and depth in scale direction at scale space. With shorter sampling distance and larger sampling depth, the extracted feature points are more robust to scale changing.
- Set a threshold $Th$, when searching local extremum in $3\times3\times3$ region of scale space, DoG function $D(x,y,\sigma)$ should meet the following formula:

$$
\left\{D(x, y, \sigma) \parallel D(x, y, \sigma) > Th\right\}
$$

(2)

In our experiences we set $Th=0.05$

The improving method mentioned above can strengthen the robustness of extracted feature points, but their distribution is still not equal. The distribution of local feature points is related to the performance of watermarking systems. In other words, the distance between adjacent feature points should be determined carefully. In order to fit our work, their number and distribution need to be adjusted. The following is adjusting method.

Detect feature points with the improved SIFT algorithm firstly. In the local circle regions centered detected feature points with the radius $R$, if the DoG function $D(x,y,\sigma)$ value of circle center is the local extremum of the circle region, we keep the feature points, otherwise remove the points. In this way, remained feature point is the most robust point in the local circle region. The distribution of remained feature points is comparatively even. And the distance of two feature points can keep between $R$ and $2R$. Fig. 1 (a) is the extracted feature points using improved SIFT algorithm, and Fig. 1 (b) is the distribution adjusted result by the above mentioned method.

The scale $s$ of features point is related to the scaling factor of the Gaussian function in the scale space, and varies with the local image characteristic. So the local
circle radius R is accordant with the scale of the
selected point, namely \( R = r \times s \). \( r \) is the adjusting parameter of the
distance between feature points. The value of \( r \) can be
used to control the number of feature points. With larger \( r \),
the distance between feature points is larger, the number
of remained feature points will be smaller, and the local
region to insert watermark will be larger. On the contrary,
\( r \) is smaller, the distance between feature points will be
shorter, remain more feature points, and the local region
to insert watermark will be smaller.

Smaller local region to insert watermark can be robust
to local attacks, but can not contain more watermark bits.
So setting the value of \( r \) should take into account both
robustness and watermark capacitance. In our experience
we set \( r \approx 5 \).

III. CROSS RATIO THEORY OF FOUR COLLINEAR POINTS

As seen in Fig. 2, \( A, B, C, D \) are four collinear points
they all locate at line \( m_1 \). Choose a direction along the
line as the positive direction (so that distances measured
in the opposite direction are treated as negative). The
cross-ratio \( r_{ABCD} \) of these four points in the given order
is defined to be:

\[
r_{ABCD} = \frac{AC \times BD}{BC \times AD}
\]  

Where \( \overline{PQ} \) denotes the signed distance between two
points P and Q for some choice of orientation.

Figure 2. The sketch map of cross-ratio

It is a standard result that the cross ratio of four
collinear points does not change under a transform.
Affine transform is the special form of projective
transform. The principle of projective transform is useful
to affine transform, so the cross ratio of four collinear
points also keep invariant under affine transform. The
parallel lines keep parallel after being operated by affine
transform, but do not keep parallel operated by projective
transform. And line is still line after being operated by
projective transform

From cross ratio is projective invariant, we can get the
following two lemmas:

**Lemma 1:** Suppose three points of four collinear ones
have been known, and the cross ratio of the four collinear
points is also given, the forth point is certain.

**Proof:** Suppose \( A, B, C \) and \( D \) are distinct collinear
points, as shown in Fig. 2. Now know \( A, B, C \) are certain,
and the cross ratio \( r_{ABCD} = \frac{CA}{CB} \cdot \frac{DA}{DB} \) is also certain. If the
forth point \( D \) is not certain, then segment \( BD \) is not
certain, from which can deduce the cross ratio
\( r_{ABCD} = \frac{CA}{CB} \cdot \frac{DA}{DB} \) is not certain. That is in contradiction
with the known facts. So the forth point \( D \) is certain.

**Lemma 2:** Suppose four lines \( a, b, c, d \) cross the same
point \( O \), intersect line \( l_1 \) at four points \( A, B, C, D \),
and intersect line \( l_2 \) at four points \( A', B', C', D' \), as shown in
Fig. 2. According to the duality principle of projective
geometry, the following equation can be got.

\[
r_{ABCD} = r_{abcd} = \frac{CA}{CB} \cdot \frac{DA}{DB}
\]  

**Proof:** According to the duality principle of projective
geometry, if \( r_{ABCD} \) is certain, then the cross ratio \( r_{abcd} \) of
line \( a, b, c \) and \( d \) is certain. The definition of \( r_{abcd} \)
is following:

\[
r_{abcd} = \frac{\sin(ca)}{\sin(cb)} \cdot \frac{\sin(da)}{\sin(db)}
\]  

Where \( \sin(ca) \) is the sine of the directed angle formed by
line \( c \) and \( a \).

The areas of triangle \( COA, COB, DOA \) and \( DOB \) can be
calculated by the following formula:

\[
\begin{align*}
S(COA) &= \frac{1}{2} \cdot h \cdot CA = \frac{1}{2} \cdot OC \cdot OA \cdot \sin(ca) \\
S(COB) &= \frac{1}{2} \cdot h \cdot CB = \frac{1}{2} \cdot OC \cdot OB \cdot \sin(cb) \\
S(DOA) &= \frac{1}{2} \cdot h \cdot DA = \frac{1}{2} \cdot OD \cdot OA \cdot \sin(da) \\
S(OAB) &= \frac{1}{2} \cdot h \cdot DB = \frac{1}{2} \cdot OD \cdot OB \cdot \sin(db)
\end{align*}
\]  

Where \( h \) is the high from point \( O \) to line \( l_1 \).

From the above four area computing formula, the
following equation can be deduced:

\[
r_{ABCD} = \frac{CA}{CB} \cdot \frac{DA}{DB} = \frac{\sin(ca) \cdot OA}{\sin(cb) \cdot OB} = \frac{\sin(da) \cdot OA}{\sin(db) \cdot OB}
\]  

Similarly we can deduce \( r_{ABCD} = r_{abcd} \)

So \( r_{ABCD} = r_{abcd} = r_{abcd} \)

Based on the Lemma 1 and Lemma 2, we choose the
watermark bits inserting points in the local quadrilateral
region decided by feature points, which guarantees watermark is invariant to projective transformation.

IV. WATERMARK INSERTING

The whole process of watermark inserting can be seen in Fig. 3, firstly extract feature points from carrier image with the improved SIFT algorithm given in part II, and adjust the number and distribution of feature points using the method in part II. Then choose the proper quadrilateral as local feature regions to insert watermark and get watermarked image.

A. Choose Local Feature Regions

This paper uses the local quadrilateral regions to insert watermark, and the apexes of quadrilateral are the chosen feature points. Use the improved SIFT algorithm to extract feature points, adjust the number and distribution and get a set of feature points P = \{p_i, i = 1, \ldots, N\}. And the local feature regions are decided by the feature points in set P.

The method to choose local feature regions is as the following: regard feature points p_i, i = 1, \ldots, N as center, in a round region with radius 4R searching three feature points, and construct a quadrilateral with point p_i. The chosen quadrilateral should be convex quadrilateral that is close to square. To meet this requirement, the internal angle of chosen quadrilateral should be smaller than 100 degree, the length of two adjacent sides should be similar, and their difference can not be larger than 30 pixels.

And the chosen quadrilateral regions maybe overlap each other. If two regions overlap (not include the two regions with same point and side), choose the one approaching square, which will help watermark inserting and detecting.

B. Choose the Inserting Locations for Watermark Bits

In every local feature region, decide the inserting location for watermark bit based on the projective invariance of cross ratio. Suppose P_1, P_2, P_3, and P_4 standing for a local feature region, as shown in Fig. 4, line P_1P_3 intersects line P_2P_4 at point O.

If extremity points of two segments P_1, P_2, P_3, and P_4 are invariant to projective transformation, their cross point O keeps invariance to projective transformation too. Because projective transformation is linear transformation in two dimensions space, line transformed by projective is still line. Two segment keep invariance, and their cross point keep invariance. According to Lemma 1 in part III, given three points of four collinear points and the cross ratio of the four collinear points, the forth point is certain.

Given n cross ratios in advance, locate n points in every segment, and control n/2 points distribute one side of cross point O, another n/2 points distribute the other side, as shown in Fig. 4. In the triangleP_1P_2O, draw lines between point P_1 and the points on segment P_2O, draw lines between point P_2 and the points on segment P_1O. According to Lemma 2 in part III the cross points of these lines keep invariance to projective transformation.

We choose the cross points (marked black dot in Fig. 4) as the inserting location for watermark bits, which will be robust projective transformation.

In order to keep the correct order of watermark bits when watermarked image destroyed by geometrical attacks, the watermark bit inserting location should arrange in a certain order. We regulate arranging the inserting location from the original side along clockwise of the internal angle at cross point O. Arranging order is along the arrowhead orientation in Fig. 4.

Use the same way to choose the watermark inserting location in the other three triangles.

C. Watermark Inserting Method

We adopt quantization modulation to modify the pixel value at chosen watermark bit inserting location. The pixel values at N watermark inserting locations are supposed to be I(x,y) j=1,2,\ldots,N, and watermark is a \{0,1\} sequence, supposed to be W(j) j=1,2,\ldots,N. I(x,y) and W(j) are one to one correspondence. If W(j) corresponding I(x,y) is 0, modulate I(x,y) to center of even interval; If W(j) corresponding I(x,y) is 1, modulate I(x,y) to center of odd interval.

Describe the quantization step as D, and the quantization interval \( \lambda \) is defined as:
\[ \lambda = \left[ \frac{I(x,y)}{D} \right] \]  
(8)

Where \([\cdot]\) mean rounding a number to the nearest integer.

In order to insert watermark bits, the pixel value of \(I(x,y)\) is modified by (9)

\[
I_w(x,y) = \begin{cases} 
(\lambda - 0.5)D & \text{if } (\lambda + W(i)) \mod 2 = 1 \\
(\lambda + W(i))D & \text{if } (\lambda + W(i)) \mod 2 = 0 
\end{cases}
\]  
(9)

In order to strengthen the robustness of watermarking system, the pixels in a 3×3 region centered the chosen inserting point are also modified. If the center inserting point is modified to correspond 0, the 8 pixels around is modified to correspond 0. If the center inserting point is modified to correspond 1, the 8 pixels around it is also modified to correspond 1.

In modifying the pixels with (9), how to choose the quantization step \(D\) is a crucial problem. The step \(D\) is a key balance factor between robustness and imperception of watermark. If choose large value for step \(D\), watermark will have good robustness, but not guarantee imperception. If choose small value for step \(D\), watermark will have good imperception, but not guarantee robustness. In many former reported paper, the step \(D\) was adjusted through repeated experimentation, to meet the requirement for Peak Signal-to-Noise (PSNR) of watermark. In this paper we deduce the quantitative relation between PSNR and step \(D\). Based on the relation, compute the step \(D\) with PSNR directly, not need repeated experimentation.

The following is PSNR definition.

\[
PSNR(I, I') = 10 \log_{10} \frac{255^2}{MSE} 
\]  
(10)

\[
MSE = \frac{1}{M} \sum_{i=1}^{N} (I_i - I'_i)^2 
\]  
(11)

Where MSE is mean square error, \(I\) is original image, \(I'\) is watermarked image. \(I(x,y)\) denotes the value of pixel \((x,y)\), \(M\) is pixel number in all local feature region to insert watermark, and \(N\) is pixel number used to insert watermark bits. In many reported paper, \(M\) means the size of image and PSNR stands for the imperception of the whole image. In this paper watermark is inserted local feature regions, not the whole image, if set \(N\) to be size of the whole image, PSNR will be small even local region changed largely. In this case, watermark has been seen and PSNR can not express the quality of watermarked image.

In local feature region the watermark inserting ratio is \(\rho = N / M\), the relation of step \(D\), inserting ratio \(\rho\) and mean square error MSE can be deduced.

We can learn from (9) that quantization errors distribute the interval \([-D, D]\) equally. So the mathematical expectation of the square quantization errors is (12).

\[
E(I(x,y) - I'(x,y))^2 = \frac{(2D)^2}{12} = \frac{D^2}{3} 
\]  
(12)

And the mathematical expectation of the mean square error MSE is (13).

\[
E(MSE) = \frac{1}{M} \sum_{i=1}^{N} E(I(x,y) - I'(x,y))^2 
\]  
(13)

\[
= \frac{N}{M} \frac{D^2}{3} = \frac{1}{3} \rho D^2 
\]  
(14)

So we can deduce the relation of step \(D\), inserting ratio \(\rho\) and mean square error MSE.

\[
D = 255 \left[ \frac{1}{3} \rho \cdot 10^{\frac{PSNR}{10}} \right]^{-0.5} 
\]  
(14)

V. WATERMARK DETECTING

For input image \(I'\), firstly extract feature points with the improved SIFT algorithm, adjust the number and distribution of feature points using the presented method in part II. Get a set of feature points \(Q = \{q_i, i = 1, \ldots, M\}\). Center feature points \(q_i, i = 1, \ldots, M\), in a round region with radius \(4R\) searching three feature points, and construct a quadrilateral with point \(q_i\). Choose the protruding quadrilaterals as watermark detecting regions. For every detecting region, searching watermark inserting points with the same method as watermark inserting, the cross ratios set is that used in watermark inserting process. For pixel \(I(x,y)\) extract watermark bit using following formula.

\[
W'(i) = \begin{cases} 
1 & \text{if } \left[ \frac{I'(x,y)}{D} \right] \mod 2 = 1 \\
0 & \text{if } \left[ \frac{I'(x,y)}{D} \right] \mod 2 = 0 
\end{cases} 
\]  
(15)

Where \([\cdot]\) means round down, \(D\) is the same quantization step used in watermark inserting.

In every chosen local feature quadrilateral the two diagonal lines divide it into four triangles, and watermark is inserted into every triangle repeatedly. In any one triangle of any two local feature quadrilateral detect watermark sequence is \(W(i)\). If the number of bits matched \(W'(i)\) and original watermark sequence \(W(i)\) is larger than the threshold \(T\), we would judge image \(I'\) exist watermark.

Determining whether there are watermarks in image \(I'\) is concern to threshold \(T\), which decide the error rate of watermark detector directly.

Two kinds of errors are possible for detector: the false-alarm probability and the miss probability. The false-alarm probability means no watermark inserted but detected while the miss probability means watermark inserted but detected having none. There is a tradeoff between these two error probabilities in selecting detector parameters. Typically, reducing one will increase the other.

The miss probability depends on success detection probability. It is difficult to evaluate the success detection probability of a watermarked bit. It depends on the attacks. So this paper chooses the threshold \(T\) only depending on the false-alarm probability.
For an unwatermarked image, the extracted (0.1) bits are assumed to be independent random variables, from which we learn the success probability of every bit is 0.5. Based on Bernoulli trials, match the watermark sequence $W'(i)$ detected from triangle region with original watermark sequence, if there are $k$ bits matched successfully, the false-alarm probability to detect watermark from one triangle can be computed with the following formula.

$$P_{\text{false}} = \sum_{k=0}^{N} \frac{(0.5)^{N-k} \cdot (N-k)!}{k!(N-k)!}$$

(16)

Where $N$ is the length of watermark sequence, $k$ is the number of bit matched successfully. $T$ is watermark detecting threshold.

Four triangles in a local feature region are all inserted watermark, if detect watermark from one triangle successfully, we judge existing watermark in the local feature region. The false-alarm probability to detect watermark from one local feature region can be computed from the following formula.

$$P_{\text{false}} = \sum_{i=1}^{4} (P_{\text{qua}})^{4-i} (1-P_{\text{qua}}) \frac{4!}{i!(4-i)!}$$

(17)

If detect watermark from two local feature region successfully, we judge existing watermark in image $I'$. Suppose inserting watermark in $m$ local feature region, the false-alarm probability to detect watermark from image $I'$ can be computed with the following formula.

$$P_{\text{false}} = \sum_{i=2}^{m} (P_{\text{qua}})^{m-i} (1-P_{\text{qua}})^{i-1} \frac{m!}{i!(m-i)!}$$

(18)

Suppose the length of watermark sequence $N=25$, there are 10 local feature quadrilateral inserting watermark, detect threshold $T=20$. From the above analysis we can learn the false-alarm probability $P_{\text{false}}=2.8\times10^{-3}$.

VI. EXPERIMENTATION RESULT AND ANALYSIS

We choose three gray images Lena, Pepper and Baboon to test our scheme, their size is $512\times512$. These images have different texture. In experimentation, the distance adjusting parameter $r=5$, the length of watermark sequence $N=25$, quantization step $D=15$, detect threshold $T=20$. There are 7 local feature regions inserting watermark in image Lena, there are 14 local feature regions inserting watermark in image Baboon, and there are 9 local feature regions inserting watermark in image Pepper.

A. The Visual Quality of Watermarked Image

Fig. 5 includes the watermarked Lena, Baboon and Pepper. Fig. 6 is the difference value images between original images and watermarked ones (the difference value is 30 times of original one). The quality of watermarked image is described by PSNR. If only concern the MSE of local feature regions, the PSNRs of watermarked Lena, Baboon and Pepper are 39.285, 39.259, and 39.345 respectively. If concern the MSE of the whole image, the PSNRs of watermarked Lena, Baboon and Pepper are 46.455, 46.455 and 46.455 respectively. For our scheme, the fist PSNR computing way is proper. The PSNR in this paper is computed with the fist method.

PSNR of watermarked image is related to quantization step $D$ and watermark inserting ratio $\rho$. The larger value of $D$ and $\rho$ will make watermark more robust, but the quality of watermarked image will fall off, and get smaller PSNR.

In section IV part C we have analyzed the relation of quantization step $D$, watermark inserting ratio $\rho$ and PSNR, and deduce the relation formula (14). Fig. 7 is the result value computed from theory and the experience, and giving their relation curve. The theory value is computed from formula (14) and the experience value is come from three test image.

As can be seen in Fig. 7, with the watermark inserting ratio $\rho$ increasing, PSNR decrease; and with the quantization step $D$ increasing, PSNR decrease. The experience value and theory value fit better. But $\rho=1$ and $\rho=0.5$, the experience value and theory value have little
difference. The main reason is the watermark insert location overlap with the inserting ratio $\rho$ increasing. These experiences validate the formula (12) is correct.

When inserting watermark, set inserting rate based on the area of chosen regions, and the quantization step $D$ is decided by PSNR of watermarked image. Set PSNR=40dB to guarantee the quality of watermarked image.

B. Test and Analyze the Robustness of Watermark

In order to test the robustness of the proposed scheme, we carried three kinds of experiments for watermarked image. We manipulated the watermarked image with no attack, general signal attacks and geometric attacks respectively, and then detected watermark from the attacked image.

Detecting results listed in Table 1 and Table 2, and compared with Tang’s method [8]. In table 1 and 2, “a/b” in “a/b” stands for the number of regions that detect watermark successfully, and “b” stands for the number of regions inserted watermark. “x” means failing to detect watermark.

As can be seen in table 1, when there is no attack to the watermarked images, the proposed scheme can detect all local feature regions inserted watermark, and the detected watermark bits matched the inserted watermark bits entirely.

Watermark detected results from attacked images by general signal process are also shown in table 1. From the experiments results in table 1 we can see the proposed

<table>
<thead>
<tr>
<th>Attacks</th>
<th>Watermarked Lena</th>
<th>Watermarked Baboon</th>
<th>Watermarked Pepper</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The proposed method</td>
<td>Tang’ method</td>
<td>The proposed method</td>
</tr>
<tr>
<td>No attack</td>
<td>7/7</td>
<td>7/8</td>
<td>14/14</td>
</tr>
<tr>
<td>Jpeg 80</td>
<td>6/7</td>
<td>7/8</td>
<td>10/14</td>
</tr>
<tr>
<td>Jpeg 70</td>
<td>4/7</td>
<td>7/8</td>
<td>3/14</td>
</tr>
<tr>
<td>Jpeg 50</td>
<td>2/7</td>
<td>7/8</td>
<td>x</td>
</tr>
<tr>
<td>Jpeg 40</td>
<td>x</td>
<td>5/8</td>
<td>x</td>
</tr>
<tr>
<td>Median filter (3×3)</td>
<td>6/7</td>
<td>x</td>
<td>12/14</td>
</tr>
<tr>
<td>Gaussian filter (3×3)</td>
<td>2/7</td>
<td>5/8</td>
<td>10/14</td>
</tr>
<tr>
<td>Additive noise (d=0.1)</td>
<td>x</td>
<td>5/8</td>
<td>x</td>
</tr>
<tr>
<td>Gaussian filter (3×3) +Jpeg90</td>
<td>2/7</td>
<td>5/8</td>
<td>10/14</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attacks</th>
<th>Watermarked Lena</th>
<th>Watermarked Baboon</th>
<th>Watermarked Pepper</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The proposed method</td>
<td>Tang’ method</td>
<td>The proposed method</td>
</tr>
<tr>
<td>Remove 5 rows and 17 columns</td>
<td>5/7</td>
<td>x</td>
<td>11/14</td>
</tr>
<tr>
<td>Remove 8 rows and 20 columns</td>
<td>4/7</td>
<td>x</td>
<td>11/14</td>
</tr>
<tr>
<td>Cropping 10%</td>
<td>2/7</td>
<td>2/8</td>
<td>10/14</td>
</tr>
<tr>
<td>Cropping 20%</td>
<td>2/7</td>
<td>x</td>
<td>8/14</td>
</tr>
<tr>
<td>Shearing-x-5%-y-5%</td>
<td>5/7</td>
<td>x</td>
<td>11/14</td>
</tr>
<tr>
<td>Shearing-x-10%-y-10%</td>
<td>3/7</td>
<td>x</td>
<td>9/14</td>
</tr>
<tr>
<td>Rotating 1°+ Cropping</td>
<td>6/7</td>
<td>3/8</td>
<td>12/14</td>
</tr>
<tr>
<td>Rotating 30°+ Cropping</td>
<td>5/7</td>
<td>x</td>
<td>8/14</td>
</tr>
<tr>
<td>Rotating 90°</td>
<td>7/7</td>
<td>x</td>
<td>14/14</td>
</tr>
<tr>
<td>Scaling 0.5</td>
<td>5/7</td>
<td>x</td>
<td>11/14</td>
</tr>
<tr>
<td>Scaling 1.2</td>
<td>6/7</td>
<td>x</td>
<td>11/14</td>
</tr>
<tr>
<td>Linear geometric</td>
<td>7/7</td>
<td>4/8</td>
<td>12/14</td>
</tr>
<tr>
<td>transformation (1.01,0.013,0.009,1.011)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project transformation</td>
<td>6/7</td>
<td>x</td>
<td>12/14</td>
</tr>
<tr>
<td>[0.05 0.02;1.2 0; 1 1; 0 1.05]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project transformation</td>
<td>5/7</td>
<td>x</td>
<td>10/14</td>
</tr>
<tr>
<td>[0.2 0.1; 1.01 0; 1 1; 0 0.9]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rotating 30°+Scaling 0.8+Jpeg 90</td>
<td>5/7</td>
<td>x</td>
<td>7/14</td>
</tr>
</tbody>
</table>
method is robust to Jpeg compression, median filter and Gaussian filter, but failed to additive noise and Jpeg compression with laguer compression rate. For 3×3 median filter attack, the proposed method shows more robust than Tang’s method. The main reason is that SIFT algorithm is more robust to median filter than Mexican hat wavelet scale interaction algorithm. Watermark bit is inserted in 3×3 region around the chosen point, which also strengthens the stability to median filter.

For the additive noise with intension 0.1, expect the watermarked Pepper (detect watermark in two local feature regions in watermarked Pepper), the other two images failed to detect watermark. The main reason is that SIFT feature points are sensitive to noise. We also seen the proposed method fails to Jpeg compression with quality factor less than 70. The main reason is that the proposed method inserts watermark into space domain and Tang’s method into DFT domain. The low frequency coefficient in frequency domain is more stable to Jpeg compression than pixel in space domain.

The geometric attack include rotation, scaling, translate, cropping, remove column and row, shearing, linear geometric transformation, projective transformation and their combination. Part experiment results are shown in table 2. Compared with Tang’s method the proposed method is robust to wider range of geometric attacks. For rotation, Tang’s method only resists the rotation under 5°, while the proposed method can resist rotation with arbitrary angle. For projective transformation, Tang’s method is no useful, but the proposed method can resist a certain degree projective transformation.

From table 2 we can see the proposed method has better performance than Tang’s method in resisting geometric attacks. There are two main reasons: (1) SIFT feature is more stable to geometric transformation than Mexican hat wavelet scale interaction feature. (2) the proposed method choose watermark inserting location based on cross ratio, which is invariant to projective transformation, making watermark robust to projective transform.

VII. CONCLUSION

Utilizing two kind of invariance, namely feature points extracted by improved SIFT algorithm and cross ratio of collinear points, this paper presents a watermarking method to resisting geometric attack. The method holds the merit of feature-point-based watermarking, and compared with Tang’s method, a classical feature-point-based watermarking method, it has stronger ability to resist geometric attack. The main contribution of this paper include: (1) combine the watermarking system, improved SIFT algorithm and get faster computing speed and more stable feature points. (2) Based on the cross ratio invariance to projective transformation, it choose the location of watermark bit, which guarantees watermark inserting and inserting location synchronization. (3) For quantization modulation embedding method, analyze the relation between quantization step and PSNR, and deduce their relation formula, which help decide the quantization step directly according to the given PSNR.

ACKNOWLEDGMENT

This work was supported in part by a grant from Science Technology Project of Henan, China (No.0624260019) and National Natural Science Foundation of China (No.092102310163 and 082400410210)

REFERENCES


Li Jing, received her PH.D. degree in computer software theory from Information Engineering University, Zhengzhou, China in 2009. Now she is an ASSOCIATE PROFESSOR of School of Computer and Information Engineering, Henan University of Economics and Law, Zhengzhou, Henan Province, China. Her research fields include artificial neural network,, data mining and image process, etc.
Dr. Jing is also a member of China Computer Federation.

Xiaowen Zhang, is a postgraduate in the department of computer science, University College Cork, Ireland.