Geometric-Invariant Image Watermarking by Key-Dependent Triangulation

Shiyan Hu
Department of Electrical and Computer Engineering
Michigan Technological University
Houghton, MI 49931, USA
E-mail: shiyan@mtu.edu

Keywords: image watermarking, geometric invariance, key-dependent triangulation, security, robustness

Received: August 9, 2006

Fast and massive dissemination of image data across the Internet imposes great challenges of protecting images against illegal access and unauthorized reproduction. Image watermarking provides a powerful solution for intellectual protection. In this paper, a new image watermarking approach robust to various geometric distortions is proposed. The new scheme involves detecting image feature points and triangulating them in a secure key-dependent manner. The neighborhood pixel ratio of gray-scale image is investigated in the paper. It is a novel robust image feature which can be seamlessly combined with the proposed key-dependent triangulation scheme. A random pre-warping framework is adopted to make the scheme robust to collusion attack. Our experiments demonstrate that the new scheme is robust to rotation, scaling, noise addition, JPEG compression, StirMark, shearing transformation, collusion, and other common attacks in both spatial and frequency domain.

Povzetek: Predstavljen je nov postopek za varovanje slik na internetu.

1 Introduction

Fast and massive dissemination of image data across the Internet imposes great challenges of protecting images against illegal access and unauthorized reproduction. As an effective and efficient solution, image watermarking superimposes a copyright message into a host image before dissemination and then unauthorized reproduction can be recognized by extracting the copyright information. Numerous image watermarking techniques have been proposed in the literature such as [1, 2, 3, 4, 5, 6, 7, 8]. Along with the rapid growth of novel watermarking schemes, various attacking attempts have also been developed to destroy watermarks. Among these attacks, geometric attacks are very difficult to handle. This is mainly due to the fact that slight geometric manipulation to the marked image, such as scaling or rotation, could significantly reduce the possibility of a successful watermark retrieval, provided that the watermarking extractor has no knowledge of the distortion parameters. In another word, geometric distortion can easily introduce synchronization errors into the watermark extracting process [9, 10]. In recent years, a number of approaches have been proposed to counteract the geometric synchronization attacks. Popular techniques in the literature can be loosely classified into three categories:

1. Geometric invariant domain based watermarking schemes. In these algorithms, Fourier-Mellin transform is incorporated into some watermarking schemes (e.g., [11, 12]) to tackle with geometric attacks such as rotation, scaling and translation. However, these algorithms are computationally inefficient, hard to implement and cannot survive aspect ratio change [13].

2. Template Matching-based watermarking schemes. In this class of algorithms, a template is embedded into the host image besides the watermark. The affine geometric distortions to the marked image can be reverted using the estimated parameters through detecting the template. After compensating geometric distortions, the watermark can be easily retrieved by the watermarking extractor. The major drawbacks of these techniques are that the template can be easily detected and removed by attackers [10, 14, 15].

3. Content-based watermarking schemes. This class of watermarking schemes achieve recovery from geometric distortion using image content. A particular interesting scheme in this category is proposed in [15] where feature points are used as a content descriptor. The method works as follows. First a set of feature points which are robust to geometric distortion are detected. These points are often near corners or edges of the image. A Delaunay triangulation is computed on the feature points and the watermark is then embedded into the resulting triangles. The above technique has two drawbacks. First, it always computes a Delaunay triangulation on the feature points. Therefore, provided that an attacker can successfully retrieve those feature points, the presence of a watermark can be easily determined and the watermark may be removed or distorted. Since usually well-known feature-point detectors (e.g., Harris detector [16]) are adopted, most feature points can actually be found by attackers. Second, the method is not as robust as expected: feature points are robust (i.e., can be completely retrieved by the watermarking extractor) only against small-degree rotation, and the per-

formance of these techniques is considerably compromised when large-degree rotation occurs. This drawback has also been reported in [15, 17].

In this paper, we propose a new feature point-based image watermarking algorithm which is secure and robust to common attacks in both spatial and frequency domain. The proposed scheme first generates four image feature points using a novel *robust intersection-based feature point detector*. Based on the feature points, a number of additional points are generated and then triangulated in a *key-dependent* manner. Finally, the watermark is embedded into the resulting triangles. The key dependance properties of the proposed technique is motivated by the following results from the computational geometry literature. It is shown in [18] that there exist $\Omega(2.33^n)$ different triangulations for a planar set of n points in general position. Therefore, even if attackers repeat the feature points, they are generally not able to compute the right triangulation. We also consider the application of the proposed watermarking scheme to image fingerprinting. Under this situation, robustness against collusion attacks becomes critical. Therefore, a random pre-warping framework is adopted to make the proposed scheme robust against such attacks.

The performance of the new scheme is substantiated by the extensive experiments. Our experimental results demonstrate that the new scheme is robust to rotation, scaling, noise addition, JPEG compression, StirMark, shearing transformation, collusion, and other common attacks in both spatial and frequency domain.

The rest of the paper is organized as follows: Section 2 describes the robust intersection-based feature point detector. Section 3 describes the key-dependent triangulation-based watermarking scheme. Section 4 presents the experimental results and analysis. A summary of work is given in Section 5.

2 Robust intersection-based feature point detector

The first step is to compute some feature points from an image. To this effect, numerous techniques can be applied, however, even the popular Harris detector [16] cannot guarantee the repeatability of feature points after a large degree rotation [15, 17]. To settle this problem, our strategy is that we first rotate an image by each integer degree, and apply Harris detector to each resulting image. The intersection of the detected points forms the feature point set. Note that smaller degree interval could be applied, however, integer degree interval suffices as indicated in our experiments. The parameters of Harris detector are determined as follows. In principle, we try to find a nice set of parameters such that the intersection of feature points from all images, after *rotated back*, contains only four feature points. Therefore, all that we need to record for the watermarking extractor is the set of these parameters and the key. The latter will be described in Section 3.

For convenience, we use the popular Lena image as an example to illustrate the ideas in this paper. We first rotate the Lena image (of size 512×512) by $1^\circ, 2^\circ, \ldots, 359^\circ$ and then apply Harris detector with the same parameter values to each resulting image. Some detection results are shown in Fig. 1. We then rotate each image back (e.g., we rotate the second image in Fig. 1 by 15° in clockwise direction), and compute the intersection of all the feature points after necessary translation. The resulting intersection contains only four points as illustrated in Fig. 2(a). For completeness, some details of our implementation of Harris detector [16] are elaborated as follows.

1. Compute x and y derivatives of image I

$$I_x = G_\sigma^x * I, I_y = G_\sigma^y * I. \tag{1}$$

2. Compute products of derivatives at every pixel

$$I_{x^2} = I_x \cdot I_x, I_{y^2} = I_y \cdot I_y, I_{xy} = I_x \cdot I_y. \tag{2}$$

3. Compute the sums of the products of derivatives at each pixel

$$S_{x^2} = G_{\sigma'} * I_{x^2}, S_{x^2} = G_{\sigma'} * I_{y^2}, S_{xy} = G_{\sigma'} * I_{xy}. \tag{3}$$

4. Define at each pixel (x, y) the matrix

$$H(x,y) = \begin{pmatrix} S_{x^2}, S_{xy} \\ S_{x,y}, S_{y^2}. \end{pmatrix} \tag{4}$$

5. Compute the response of the detector at each pixel

$$R = Det(H) - k(Trace(H))^2. \tag{5}$$

Several parameters are to be determined: the sigma of Gaussian derivatives, the sigma of the Gaussian integration, the k in the computation of "cornerness", the size of the window for computing the local maximum in R, and finally the threshold for "cornerness". The parameters used for Fig. 1 are $\sigma = 0.5, \sigma' = 0.8, k = 0.05, Theshold = 52000$, and window size is set to 30 × 30.

The heuristic to determine the parameters reads as follows. σ, σ', k and window size are first set and $Theshold$ is changed from larger values to smaller values to obtain the desired effect. It is possible that Harris detector with a set of parameters generate more than four intersection points, while slightly increasing the threshold will lead to less than four intersection points. In this case, we do not increase the threshold, instead, we compute four special points from the obtained intersection points. The following process is also useful for recomputing the intersection points in breaking a tie (see Section 3.1). Suppose that there are k intersection points. We first compute the convex hull [19] on them and three cases follow.

(1). Exactly four points lie on the convex hull. Then they are returned as the final intersection points.

(2). More than four points lie on the convex hull. In this case, the hull edges are sorted according to their lengths and points p_i and p_j linking the shortest edge are merged. That is, p_i (resp. p_j) is removed if p_i (resp. p_j) is to the left

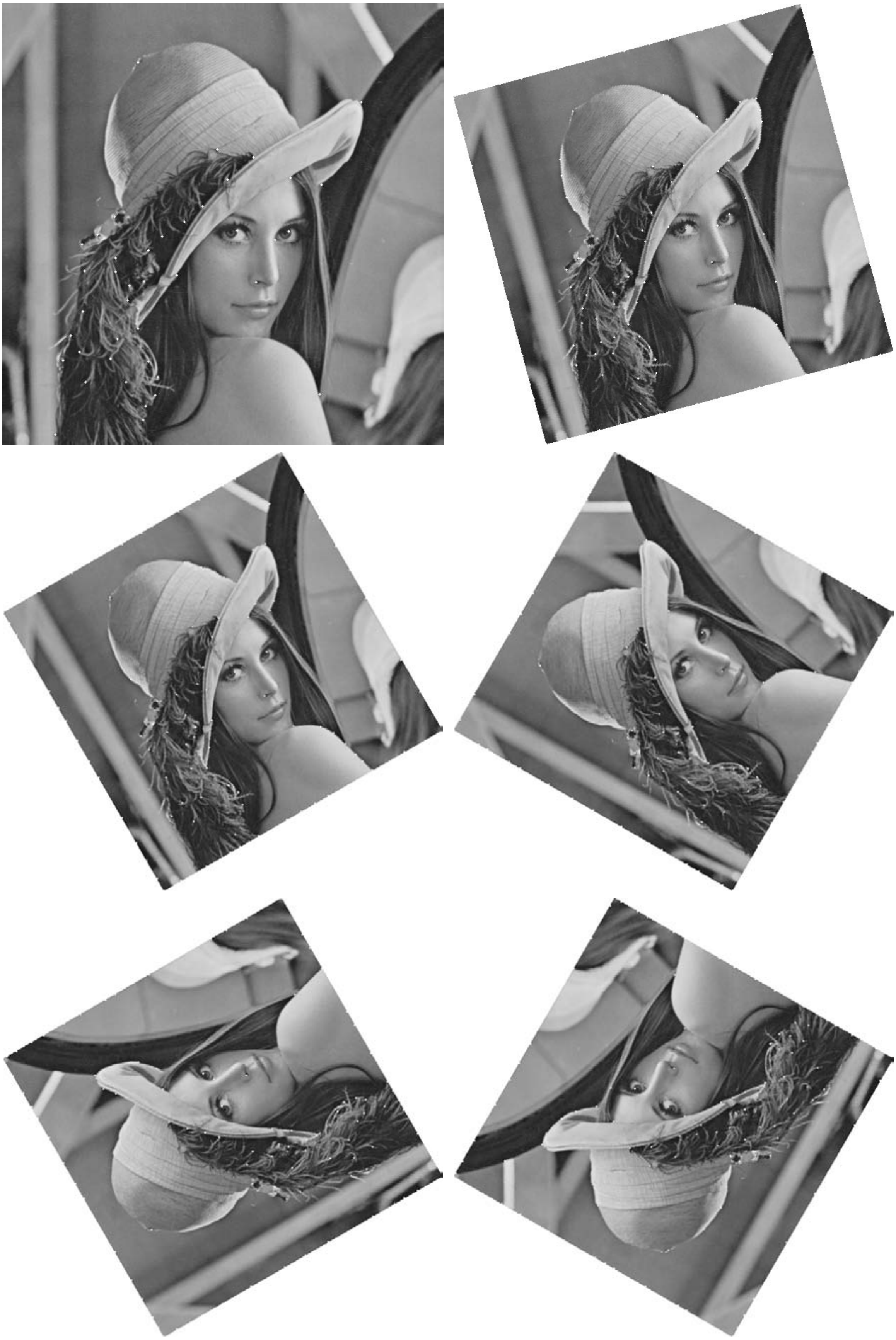

Figure 1: Feature points (denoted by $+$) obtained by Harris detector for Lena after rotation of $0^\circ, 15^\circ, 30^\circ, 60^\circ, 120^\circ, 150^\circ$, respectively.

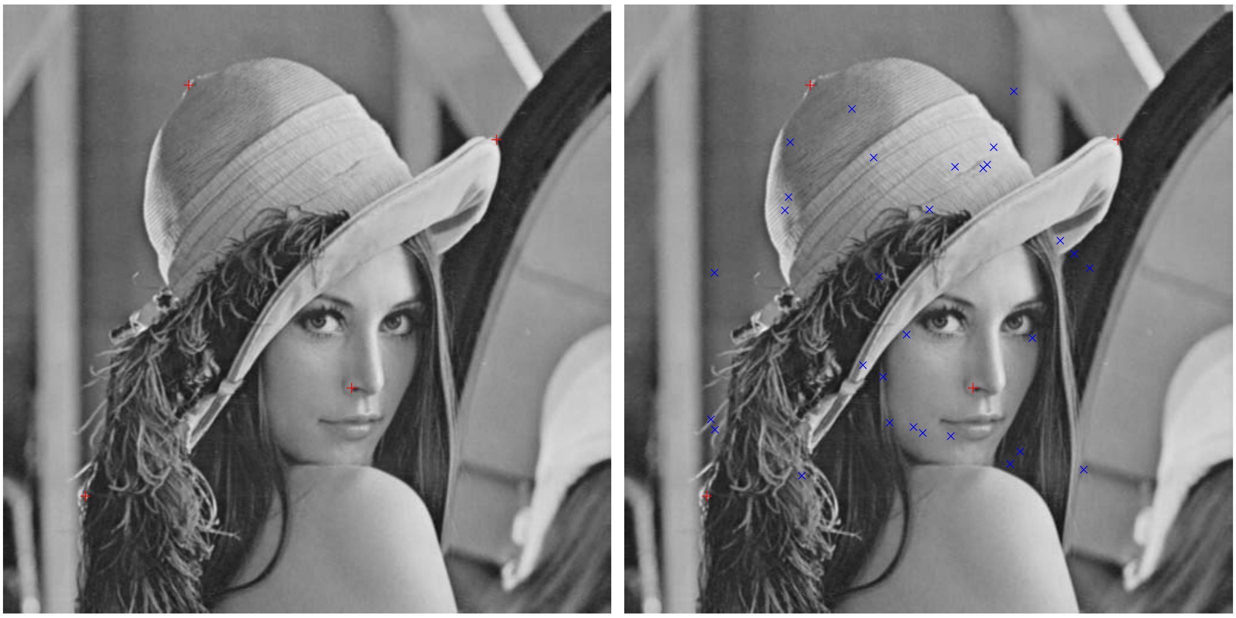

Figure 2: (a) Intersection points: + denotes the intersection of feature points by Harris detector for Lena. (b) New points for Lena: + denotes four feature points, × denotes 30 generated points.

of p_j (resp. p_i) in clockwise direction, and the convex hull is then accordingly updated. The process is repeated until only four points are on the convex hull.
(3). Three points lie on the convex hull. We then arbitrarily pick a feature point inside the hull and return these four points. The newly picked point does not impact the generation of additional points, refer to Section 3.1.

3 Key-dependent triangulation based watermarking

3.1 Generating additional points

Through the above phase, we have four feature points in hand. The following process is carried out to generate N new points in a key-dependent manner. Key-dependent property involves using pseudo-random numbers. Throughout this paper, pseudo-random numbers are generated depending on a secret key, which is stored for the watermarking extractor. The procedure reads as follows.
(1). In the case of four hull vertices, we first compute the longer one of the two *diagonal* segments formed by these vertices. Denote by p_a, p_c the two endpoints and by p_b, p_d other points where p_a, p_b, p_c, p_d are in clock-wise order and $p_a = \text{argmax}_{p_i \in \{p_a, p_c\}} d_2(p_i, p_b) + d_2(p_i, p_d)$, $d_2(\cdot)$ being the Euclidean distance function. In the case of three hull vertices, the longest hull edge is returned as $p_a p_c$ and another hull vertex is p_b such that p_a, p_b, p_c are in clock-wise order. p_d is the point inside the hull. Rotate the image such that $\overrightarrow{p_c p_a}$ is 45° with respect to the horizontal direction. Whenever there is a tie, we choose another set of parameters for Harris detector to recompute the intersection points. Note that images shown in this paper are first rotated back to its original position for the convenience of illustration.
(2). Compute the bounding box of feature points in the rotated image as follows. Let $A = \{a, b, c, d\}$, and let $minx = \min_{i \in A}\{x_i\}, maxx = \max_{i \in A}\{x_i\}, miny = \min_{i \in A}\{y_i\}, maxy = \max_{i \in A}\{y_i\}$. The bounding box is defined by $\{(minx, miny), (maxx, miny), (maxx, maxy), (minx, maxy)\}$.
(3). Generate two uniform deviates λ_1 and λ_2 with our key. A new point is generated as $(minx \cdot \lambda_1 + maxx \cdot (1 - \lambda_1), miny \cdot \lambda_2 + maxy \cdot (1 - \lambda_2))$.
(4). Repeat Step (3) until the number of new points reaches N.

Refer to Fig. 2(b) for 30 newly generated points. Taking the first generated point as the *origin* and the vector formed by the first and the second points as the direction of x-coordinate, we build up a *reference coordinate system* conforming to the right-hand rule. Note that the original four feature points will not be used in the following watermark embedding process. We are now ready to triangulate the N new points in a key-dependent way.

3.2 Computing key-dependent triangulation

Refer to Fig. 3(a) for a reference coordinate system with the origin v_s, where the dotted line with an arrow represents x-axis. Recall that a *triangulation* of a planar point set is a maximal set of non-intersecting straight-line segments connecting points in it [19]. The key-dependent triangulation is computed as follows.
(1). Sort all vertices (i.e., points) by their polar angle in the reference system. The distance to the origin v_s is used

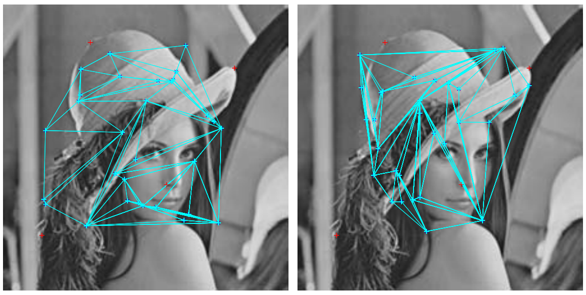

Figure 4: Sample triangulations for Lena with different generated points.

for tie breaking. Denote the resulting set by V^1. Refer to Fig. 3(b) where the vertices are numbered in the order.

(2). The triangulation is built incrementally and involves using pseudo-random numbers. First compute as V_{v_s} the set of all vertices visible to v_s in V. Note that v_a and v_b are visible to each other if v_a, v_b are distinct and the line segment v_av_b does not intersect with any existing segment in the incomplete triangulation. Of course, we require that v_av_b itself has not been inserted yet. In Fig. 3(b), there is no inserted segment, so $V_{v_s} = \{v_1, \ldots, v_8\}$. Randomly pick a vertex v_{h_1} from V_{v_s} and insert line segment $v_sv_{h_1}$. We then compute $V_{v_{h_1}}$ which contains all the vertices visible to v_{h_1}, randomly select a vertex v_{h_2} from $V_{v_{h_1}}$, and insert the segment $v_{h_1}v_{h_2}$ into the graph. As an illustration, suppose that at a time point, the incomplete triangulation is as Fig. 3(c) and the current vertex is v_7, then $V_{v_7} = \{v_4, v_6, v_8\}$. The above process is repeated until either the triangulation is completed or no vertex is visible to the current vertex. Suppose that the triangulation proceeds to Fig. 3(d). The last few visited points form the sequence of $v_8 - v_1 - v_2 - v_3 - v_5 - v_4$. Since there is no vertex visible to the current vertex v_4, we have to find the next vertex through backtracking, i.e., the current vertex is changed to v_5 which is the most recent vertex except v_4. If yet no visible vertex for v_5, we will continue the backtracking process. In the case of Fig. 3(d), we need to backtrack to v_3 where $V_{v_3} = \{v_s, v_1, v_8\}$. A complete triangulation is shown in Fig. 3(e). For the Lena image, two sample triangulations are shown in Fig. 4.

3.3 Watermark embedding process

Suppose that the reference coordinate system and the triangulation can be repeated after various attacks to the original image, the present issue is to come up with a robust embedding method for each triangle. For this purpose, we investigate as follows a neighborhood information-based image feature which is robust to various attacks such as noise, JPEG compression and geometric distortions. This feature has been successfully applied to watermarking binary document images in the previous work [20]. However, more work is needed for extending this tool to watermark grayscale images.

First note that too small triangles (with the area below a threshold) are first eliminated from consideration. All remaining triangles are then ordered/indexed in the following way. For two triangles defined by vertices v_a, v_b, v_c and v_d, v_e, v_f respectively (without loss of generality, assume that $a < b < c$, $d < e < f$), the order of them is determined by a and d; If $a = d$, then compare b and e; If still tie, then compare c and f. Given N points, all triangles are uniquely indexed from 1 to $N' \leq 2N - k - 2$ where k is the number of vertices on the convex hull of these N points (see [19]). Note that the inequality is due to removal of small triangles. We call each triangle *a partition* of the image. Denote each partition as par_i, $i = 1, \ldots, N'$.

We now discuss how to embed the watermark bitstream to a single partition. The *weight* $w(par_i)$ of a partition par_i is defined as follows. A counter, initialized to 0, is associated to par_i. For each pixel g inside par_i, we check for its eight neighbors: if more than three neighbors of g have intensity values larger than a threshold T_L, the counter corresponding to par_i is incremented. $w(par_i)$ is defined as $\frac{counter}{area_i}$ where $area_i$ denotes the area in pixels of par_i. We call this ratio (or *partition weight*) the *neighborhood pixel ratio* (NPR). The NPR ratio for gray-scale image is an extension of the NPR ratio for binary image, which is a robust feature as shown in [20].

For a single partition, noise can be often “filtered out”, e.g., random noise can be “filtered out” from incrementing the counter since we increase it only when at least four neighbors of a pixel have intensity values larger than T_L. Such a computation captures the intrinsic characteristic of a partition to some extent. We illustrate this fact through an example, refer to Fig. 5(a) for a partition from Lena. The involved threshold T_L is set to 120.

The original partition Fig. 5(a) and the noisy partition Fig. 5(b) are visually different, however, their neighborhood pixel ratios are 0.3267 and 0.3140, respectively. Two ratios differ only by 1%! We also test NPR ratio for scaling, rotation, JPEG compression, low pass and median filtering. Refer to Table 1 for the resulting NPR ratios. All ratios are similar to each other. In contrast, the ratio of another partition from Lena shown in Fig. 5(d) computes to 0.2183, which differs from the original’s by 31%! The combinations of the above attacks are also performed, and the NPR ratio is resistant to these attacks. As an example, Fig. 5(c) shows a rotated noisy partition whose NPR ratio computes to 0.3129, very close to 0.3267. The above experiments demonstrate the robustness of the neighborhood pixel ratio. Refer to Section 4 for further experiments on robustness against geometric distortions.

It remains to present the principle for modifying a partition - we only modify pixels whose intensity values are close to T_L. For each such pixel, we may either increase or decrease the intensity value depending on whether the ratio is to be increased or decreased. For example, increasing a pixel’s intensity from $T_L - 2$ to T_L is *possible* to increase the partition weight. The process is repeated un-

[1] $v_s \in V$ may be regarded as the 0-th vertex.

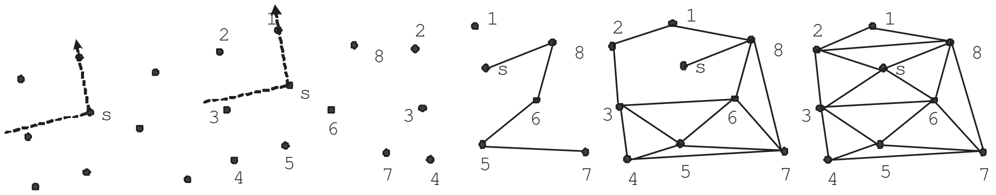

Figure 3: A simple example for key-dependent triangulation, from left to right (a)(b)(c)(d)(e).

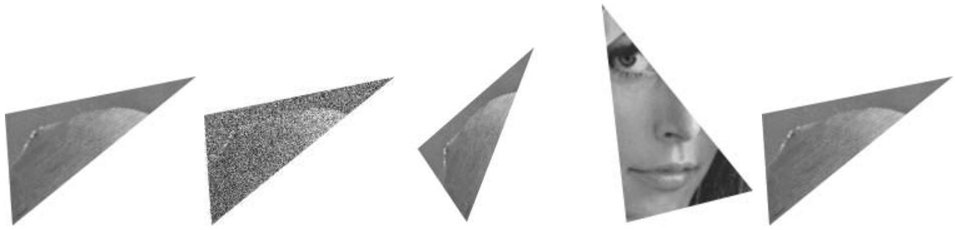

Figure 5: Top to bottom, left to right: (a) the original partition (b) the partition corrupted with synthetic Gaussian noise with $\sigma = 30$, (c) rotating by 30° followed by noise addition (the resulting image is scaled down here due to space limitation) (d) another partition (e) a modified partition of (a).

Table 1: NPR ratio for the attacked image partition.

Attack	A partition with the weight of 0.3267
JPEG with QF of 10%	0.3178
Add. White. Gauss. Noise, $\sigma = 20$	0.3140
Low Pass Filter	0.3126
3×3 Median Filter	0.3098
Scale by Factor of 0.5	0.3248
Scale by Factor of 2.0	0.3263
Rotation by $30°$	0.3202
Rotation by $60°$	0.3126

til the counter reaches the goal within an error of $4/area$. For instance, refer to Fig. 5(e) for a modified partition (of Fig. 5(a)) whose ratio is 0.2651 compared to the original ratio 0.3267. The modification is not visually perceptible.

Based on the NPR ratio, we are ready to present the watermark embedding process, which is motivated by [21]. Recall that every triangle is indexed. We first randomly select $\lfloor N'/2 \rfloor$ triangles and denote the triangle sequence by $O_1, O_2, \ldots, O_{\lfloor N'/2 \rfloor}$. We then randomly select the remaining triangles to form the sequence $Z_1, Z_2, \ldots, Z_{\lfloor N'/2 \rfloor}$.

The watermarking extractor works to retrieve the embedded bitstream. It compares $w(O_i)$ to $w(Z_i)$ for each i to decide about the marked bit: if $w(O_i) - w(Z_i) \geq T_J$, then a 1 is embedded; if $w(Z_i) - w(O_i) \geq T_J$, then a 0 is embedded. Therefore, the watermarking embedder needs to accordingly modify the relationship between $w(O_i)$ and $w(Z_i)$ to embed bitstream. To this effect, without loss of generality, assume that we aim to embed a 0, however, presently $w(O_i) > w(Z_i)$. In this case, we modify the partitions to change $w(O_i)$ to $\frac{w(O_i)+w(Z_i)}{2} - \frac{T_J}{2}$ and $w(Z_i)$ to $\frac{w(O_i)+w(Z_i)}{2} + \frac{T_J}{2}$.

3.4 Robustness against collusion attacks

In this section, we discuss the application of the proposed watermarking algorithm to image fingerprinting. The term *fingerprinting* refers to superimposing a unique watermark onto each copy of the distributed data. The embedded watermark can be used to identify the unauthorized copies of the data [22]. Digital watermarking can naturally serve as an effective and efficient approach to fingerprint digital data. However, a main shortcoming in applying conventional image watermarking techniques for fingerprinting is that they are not designed to be robust against *collusion* attacks, which are common attacks to destroy fingerprints. These attacks are performed by a group of colluders with the same digital data containing different fingerprints. A common implementation of such an attack is simply averaging multiple marked versions of an image [23]. Most existing watermarking schemes robust to collusion, such as [24, 25, 26, 27, 28, 29], have shortcomings including compromised watermarking capacity and decreasing effectiveness with the increasing number of colluders [22].

In this paper, we adopt the random pre-warping framework originally proposed in [22] to design a collusion and geometric resistant watermarking scheme. For completeness, some details of the approach in [22] are included as follows. Other than trying to detect collusion and identify traitors by fingerprint, the random pre-warping framework shoots for preventing traitors from obtaining a high-quality copy through collusion. Basically, the method randomly distorts the host image *before* embedding fingerprint to it such that averaging multiple versions will introduce annoying artifacts and only result in a low-quality image with no commercial value. The idea is feasible due to the following reasons as shown in [22]. For additive watermarking procedures, an averaged image from K distinct copies can be represented as [22]

$$S_a = \frac{1}{K}\sum_{i=1}^{K} S_i = S + \frac{1}{K}\sum_{i=1}^{K} W_i, \tag{6}$$

where S is the host image, W_i is the watermark, and $S_i = S + W_i$ is the watermarked image. It is expected that S_a looks similar to S due to the fact that $\frac{1}{K}\sum_{i=1}^{K} W_i$ should vanish since each watermark W_i can be regarded as a random pattern. In contrast, if we distort S before superimposing W_i onto it, we have [22]

$$S_a = \frac{1}{K}\sum_{i=1}^{K} S_i = \frac{1}{K}\sum_{i=1}^{K} \phi_i(S) + \frac{1}{K}\sum_{i=1}^{K} W_i, \tag{7}$$

where $\phi(\cdot)$ denotes the distortion function. Even if the second term vanishes, we can choose $\phi(\cdot)$ such that $\frac{1}{K}\sum_{i=1}^{K} \phi_i(S)$ is visually different from S citeCST04. It is shown in [22] that this can be achieved using the standard Stirmark tool [30, 31] which forms the basis of desynchronization attacks on many watermarking schemes.

With the above introduction, a watermarking scheme robust to both geometric attacks and collusion attacks is clear: we first randomly pre-warp the image followed by embedding watermarks as in the previous sections.

The complete process for embedding watermarks onto a host image is summarized in Algorithm 1. To extract a watermark from a possibly modified marked image, we carry out the extracting process as shown in Algorithm 2.

4 Experimental results

We have performed experiments over various gray-scale images. We choose the Lena image to present our results and analysis. The extensive experiments on our whole image set are described in the end of this section (see Table 3).

To evaluate the robustness of the proposed watermarking scheme, common attacks are tested. For a possibly modified watermarked triangulation, we define the watermark strength as follows. Recall that $w(Z_i) - w(O_i) \geq T_J$ denotes embedding 0 while $w(O_i) - w(Z_i) \geq T_J$ denotes embedding 1. When a marked triangle is attacked, a 1 (resp. 0) can be extracted if $w(O_i) - w(Z_i) > 0$ (resp. $w(O_i) - w(Z_i) < 0$). Therefore, our scheme can tolerate up to T_J unit changes in triangle weight. Denote by ξ_1 (resp. ξ_2) the smallest value of $w(O_i) - w(Z_i)$ (resp. $w(Z_i) - w(O_i)$) for all i where a 1 (resp. 0) is actually embedded. The *watermark strength* is defined as $\frac{\min\{\xi_1, \xi_2\}}{T_J}$. Clearly, the watermark becomes more robust with the increasing value of the watermark strength. It follows from Section 3.3 that the maximum possible value of watermark strength is 1[2]. If a watermark strength is negative, the watermark may not be correctly extracted. When this hap-

[2] Exception occurs when relationship between $w(Z_i)$ and $w(O_i)$ for each i exactly matches the embedding bit sequence. However, it is very unlikely and not observed in the experiments.

Algorithm 1 Watermark Embedding Process

```
1: Use Stirmark to randomly pre-warp the image. Only geometric distortions in StirMark is applied.
2: Compute the four feature points as in Section 2, i.e., determine a set of parameters such that the intersection of feature points for all rotated image versions contains only four points. The set of parameters is recorded for the watermarking extractor.
3: Based on the intersection points, generate $N$ new points in a key-dependent manner. The key is recorded for the watermarking extractor.
4: Triangulate the generated points in a key-dependent manner.
5: Remove too small triangles whose areas are below a threshold and index the remaining ones.
6: Use two pseudo-random triangle sequences to embed the watermark as in Section 3.3.
```

Algorithm 2 Watermark Extracting Process

```
1: Use the set of recorded parameters to compute the four intersection feature points, i.e., rotate the possibly modified marked image and the intersection of feature points for all rotated image versions should contain only four points.
2: Based on the computed four feature points, generate $N$ new points using the user's key.
3: Triangulate the generated $N$ points using the user's key.
4: Remove too small triangles and index the remaining ones.
5: Generate two pseudo-random triangle sequences with the key and extract the watermark as follows. Compare $w(O_i)$ with $w(Z_i)$ for each $i$: if $w(O_i) - w(Z_i) \geq T_J$, then a 1 is embedded; if $w(Z_i) - w(O_i) \geq T_J$, then a 0 is embedded. We set up a small positive value $\delta$ for fault-tolerance purpose, i.e., we treat $T_J \approx T_J \pm \delta$.
```

pens, we are yet interested in how many bits can be correctly extracted. We are now ready to present our experimental results.

1. The different image transformations tested are scaling, rotation, and a combination of these transformations. The original Lena image of size 512×512 is shown in Fig. 6(a). The watermarked Lena image is shown in Fig. 6(b). The involved T_L is set to the average intensity value of each image. PSNR for the watermarked image is low (18.82) due to the random pre-warping by StirMark. Without applying random pre-warping, the PSNR for watermarked image is 46.13 (see Fig. 6(c)). In the case of single transformation, the proposed technique is robust to any degree of rotation (without cropping) and scaling with factor of 0.5 and 2, respectively. In the above attacks, all embedded bits are extracted. Note that in scaling attacks, all generated points lie in the bounding box of the rotated convex hull and are dependent on some random ratios (i.e., λ_1, λ_2). Therefore, as long as the scaling transformation is uniform to the image and the index (i.e., the order) of bounding box vertices is repeatable, the generated points and thus the triangulation are repeatable.

In the case of combined transformation, we test the scheme on various sequences of transformations, e.g., Fig. 6(d) shows the resulting image after the sequence of transformations including scaling by factor of 2, 20° rotation and cropping that maintains 60% of the image. All bits are successfully extracted in this class of tests, except those with too much cropping such that feature points have been removed. Since feature points are intrinsic to an image, we consider that such an attack has degraded the quality of the original image to a significant extent and thus the cropped image becomes less useful.

2. For further analysis, we have tested the proposed watermarking scheme for nonlinear geometric attacks through StirMark (with default parameter values), shearing transformation (Gimp software [32] is used to perform this transformation), compression attacks using JPEG (with a wide variety of quality factors ranging from 10% to 90%), and addition of Gaussian noise with $\sigma = 20$. In all cases, successful retrieval of watermark is reported in our experiments. Note that applying StirMark to a watermarked image still gives acceptable quality of image, see Fig. 6(e). Refer to Fig. 6(f) for the image after shearing distortion.

3. To show the robustness of the proposed scheme to re-watermarking, it is necessary to consider the following scenario. We first embed a watermark W_x to an image, then another watermark W_y is embedded to the marked image. We also embed W_y directly to the original image. Two resulting images must be different, and this is the case as demonstrated by our experiment. In addition, we even re-watermark a marked image (1) without applying the StirMark (since it usually causes considerable difference) (2) using the same set of parameters for detecting feature points. That is, two embedding processes only differ in the generated points and thus the triangulations, both of which are purely dependent on the user's key. Refer to

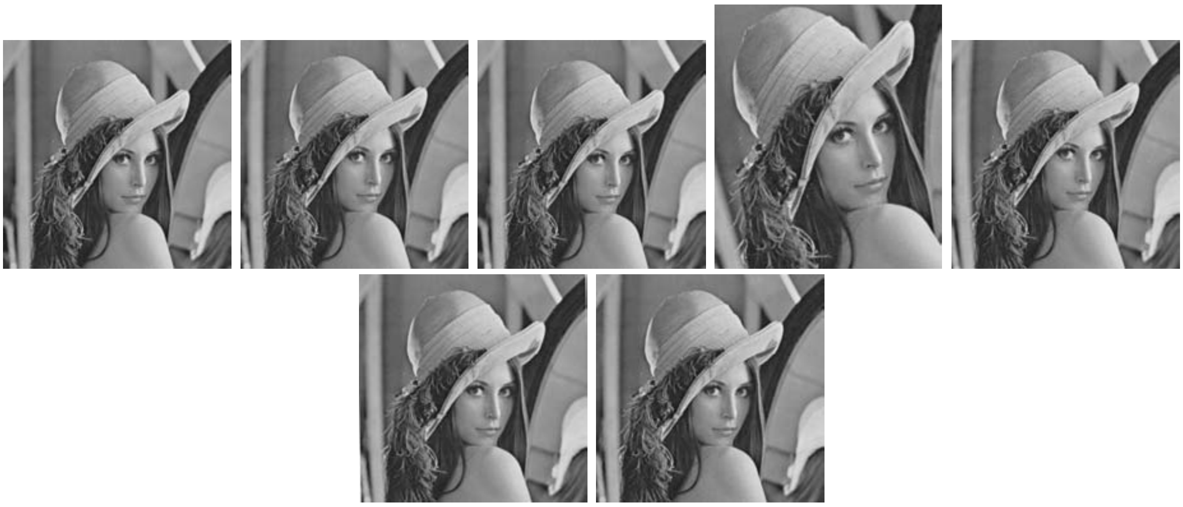

Figure 6: Left to Right, Top to bottom: (a) original Lena image, (b) watermarked Lena with random pre-warping, (c) watermarked Lena without random pre-warping, (d) scaled, rotated and cropped image, (e) StirMarked watermarked image, (f) marked image after shearing distortion, (g) re-watermarking of marked image.

Fig. 6(g) for a re-watermarked Lena over Fig. 6(b). The normalized L_1 distance between Fig. 6(b) and Fig. 6(g) is 4.02, i.e., on average the corresponding intensity values in the two images differ by 4.

4. The next test is to modify the intensity values of the marked image by either increasing or decreasing intensity value by a fixed small amount. Recall that the involved average intensity value T_L is set to the average intensity value of the whole image, thus the watermark is robust to such attacks (refer to Table 2). It is still the case for modifying intensity value by some small random amounts. A more effective attack is to modify the intensity value very close to the average intensity value. However, this attack will not really cause trouble, since instead of setting T_L to the average intensity value, we can set T_L to be the average intensity value multiplied by a key-dependent random number between $[0.5, 1.5]$.

The improved watermarking scheme by key-dependent T_L is tested and with $\delta = T_J/2$, we are *usually* able to extract all bits. The exception occurs when the attacker correctly guesses T_L and randomly exchanges considerable amount of pixels across T_L. In that case, we are still able to extract more than 80% bits as indicated by our experiments. Assume that the maximum amount for modifying any intensity value is 3, then to effectively defeat the watermarking scheme, one needs to correctly estimate the average intensity value as within the range of $[T_L - 3, T_L + 3]$. For an image with the average intensity value of 120, T_L may be of any value in [60,180]. Thus, an effective guess happens only with a probability of $6/120 = 5\%$. Refer to Table 2 for quantitative results. Refer to Fig. 7 for watermark strength versus trials of attacks. Out of 100 attacks, only 6 attacks effectively defeat the scheme.

5. Spatial domain filtering is also a class of common attacks. In our experiment, 3×3 median filtering and 3×3 mean filtering are considered. A frequency-domain low pass filter is also applied. Our further analysis includes a random change in the image's frequency domain. The maximum possible change to amplitude of FFT coefficients is $\pm 10\%$ of the original value. Refer to Table 2 for the watermark strength and the ratio of correctly detected bits after these attacks.

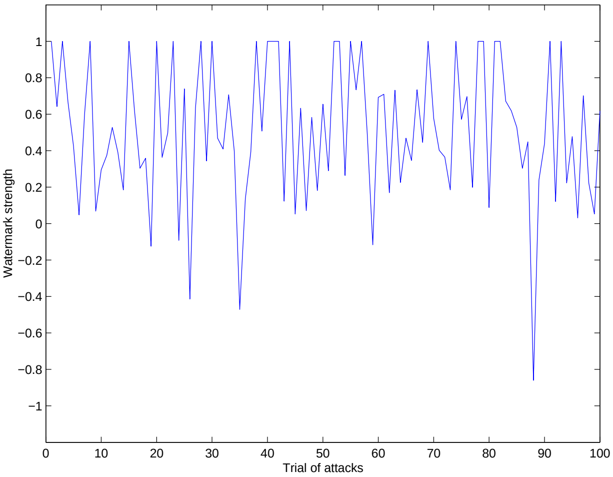

Figure 7: Watermark strength in different trials of attacks through random intensity manipulation in spatial domain.

6. Printing and Scanning. The print and scan test combines multiple attacks. For example, printing introduces re-quantization and grid apparition while scanning introduces geometric distortions [15]. Our experiments (refer to Table 2) demonstrate that the proposed scheme is robust to the print/scan attack.

7. Collusion Attacks. Due to the random pre-warping framework, the quality of colluded images should be significantly degraded. This is the case as indicated by our experiment. Refer to Fig. 8 for colluded copies of water-

marked images. Both Lenas are significantly blurred and are thus of less commercial value.

In concluding this section, we present the results for carrying out all the above tests on 50 collected images, refer to Table 3. All ratios shown are averaged over 50 images. Note that for the combinational attack with cropping, the watermark strength is not computed since some triangulations are not repeatable due to too much cropping and thus no ratio can be computed in those cases. From Table 3, one sees that the watermark strength is high for all types of attacks and on average, more than 80% marked bits can be correctly extracted even for the most effective attacks. Our experimental results clearly demonstrate the effectiveness of the proposed method.

5 Conclusion

We propose a new content-based image watermarking scheme. The scheme belongs to the class of second generation watermarking schemes whose advantages include automatic re-synchronization and exclusion of unreliable template embedding [15, 33]. The strength of the proposed scheme is demonstrated through successful watermark detection after various common attacks such as geometric distortions, StirMark attacks and shearing transformations. The main contribution of this paper is three-fold. First, a spatial domain key-dependent triangulation framework is proposed. Based on the framework, a highly secure and robust image watermarking scheme is presented. Second, a novel feature for gray-scale images, the neighborhood pixel ratio is investigated in this paper. It is an extension of the binary image NPR ratio presented in our previous work [20]. Third, detecting rotation-invariant feature points through inspecting all rotated images is investigated in this paper. Such an idea may have its own interest as well.

The proposed key-dependent triangulation framework can be easily combined with other watermarking techniques to obtain a highly secure watermarking scheme. For example, NPR ratio-based embedding process (for each triangle) could be substituted by proper existing geometric-resistant techniques. To design and analyze the combination of the key-dependent triangulation framework with the existing watermarking methods would be an interesting future work.

Acknowledgement

The author would like to thank the anonymous reviewers for their helpful comments.

References

[1] I. Cox, J. Kilian, F. Leighton, and T. Shamoon, “Secure spread spectrum watermarking for multimedia,” *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673 – 1687, 1997.

[2] J. Hernandez, M. Amado, and F. Perez-Gonzalez, “Dct-domain watermarking techniques for still images: detector performance analysis and a new structure,” *IEEE Transactions on Image Processing*, vol. 9, no. 1, pp. 55 – 68, 2000.

[3] Z.-M. Lu, D.-G. Xu, and S.-H. Sun, “Multipurpose image watermarking algorithm based on multistage vector quantization,” *IEEE Transactions on Image Processing*, vol. 14, no. 5, pp. 822 – 831, 2005.

[4] S.-H. Wang and Y.-P. Lin, “Wavelet tree quantization for copyright protection watermarking,” *IEEE Transactions on Image Processing*, vol. 13, no. 2, pp. 154 – 165, 2004.

[5] C.-H. Chang, Z. Ye, and M. Zhang, “Fuzzy-art based adaptive digital watermarking scheme,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 1, pp. 65 – 81, 2005.

[6] J. Cannons and P. Moulin, “Design and statistical analysis of a hash-aided image watermarking system,” *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1393 – 1408, 2004.

[7] A. Alattar, “Reversible watermark using the difference expansion of a generalized integer transform,” *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147 – 1156, 2004.

[8] J. Tzeng, W.-L. Hwang, and I.-L. Chern, “An asymmetric subspace watermarking method for copyright protection,” *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 784 – 792, 2005.

[9] M. Alghoniemy and A. Tewfik, “Geometric invariance in image watermarking,” *IEEE Transactions on Image Processing*, vol. 13, no. 2, 2004.

[10] P. Dong and N. Galatsanos, “Affine transformation resistant watermarking based on image normalization,” *Proceedings of International Conference on Image Processing (ICIP)*, vol. 3, pp. 489 – 492, 2002.

[11] J. Ruanaidh and T. Pun, “Rotation, scale and translation invariant digital image watermarking,” *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 1997.

[12] C.-Y. Lin, M. Wu, J. Bloom, M. Miller, I. Cox, and Y.-M. Lui, “Rotation, scale, and translation resilient public watermarking for images,” *IEEE Transactions on Image Processing*, vol. 10, no. 5, pp. 767–782, 2001.

[13] X. Qi and J. Qi, “Improved affine resistant watermarking by using robust templates,” *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. III, pp. 405–408, 2004.

Table 2: Watermark strength of the attacked Lena image.

Image	Lena	
Attack	Watermark strength	Ratio of cor. det. bits
Scale by Factor 0.5	0.95	100%
Scale by Factor 2.0	0.98	100%
Rotation by 10°	1	100%
Rotation by 20°	0.98	100%
StirMark	0.76	100%
Shearing	0.82	100%
JPEG QF 10%	0.91	100%
Noise	0.83	100%
Low Pass Filter	0.80	100%
3 × 3 Median Filter	0.91	100%
3 × 3 Mean Filter	0.82	100%
Rand. change by same amount	0.91	100%
Worst result in Rand. change in Spa. Domain	-0.25	87%
Worst result in Rand. change in Fre. Domain	0.18	100%
Print and Scan	0.32	100%

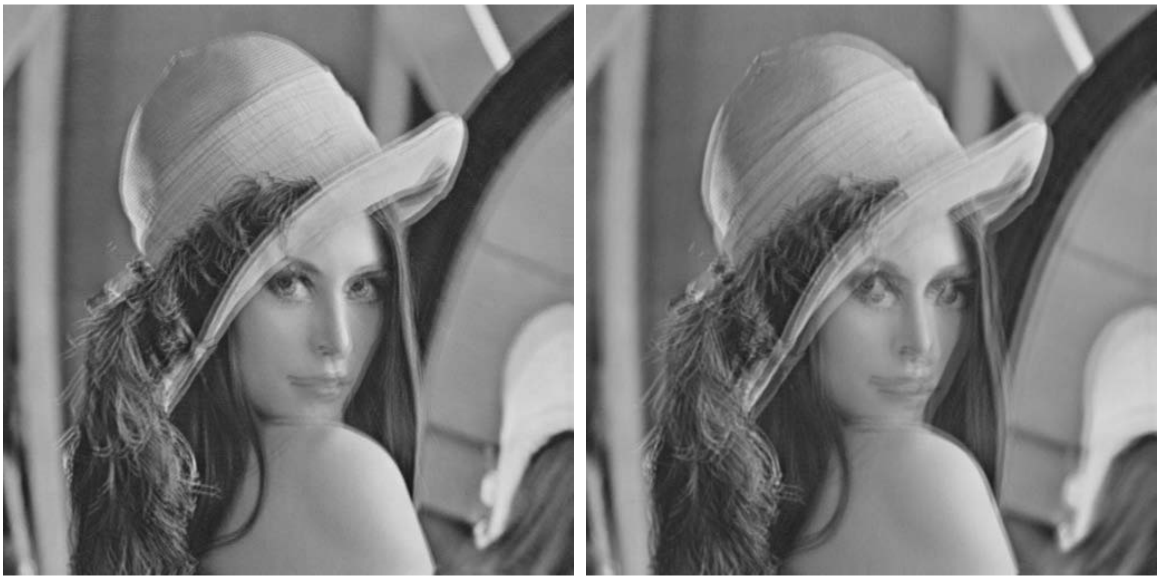

Figure 8: Colluded copies of Lena. (a) colluded image using two copies (b) colluded image using five copies.

Table 3: Averaged watermark strength over 50 attacked images.

Attack	Avg. watermark str.	Ratio of cor. det. bits
Scale by Factor 0.5	0.95	100%
Scale by Factor 2.0	0.97	100%
Rotation by 10°	0.99	100%
Rotation by 20°	0.96	100%
Rotation by 60° (no cropping)	0.93	100%
Combinational attack (no cropping)	0.95	100%
Combinational attack (with cropping)	-	83%
StirMark	0.81	98%
Shearing	0.83	98%
JPEG QF 10%	0.89	100%
Noise	0.86	99%
Low Pass Filter	0.81	100%
3 × 3 Median Filter	0.91	100%
3 × 3 Mean Filter	0.85	100%
Rand. change in Spa. Domain	0.58	89%
Rand. change in Fre. Domain	0.70	96%
Print and Scan	0.65	96%

[14] A. Herrigel, S. Voloshynovskiy, and Y. Rytsar, “The watermark template attack,” *Proceedings of SPIE: Security and Watermarking of Multimedia Contents III*, pp. 394–405, 2001.

[15] P. Bas, J.-M. Chassery, and B. Macq, “Geometrically invariant watermarking using feature point,” *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 1014–1028, 2002.

[16] C. Harris and M. Stephens, “A combined corner and edge detector,” *Proceedings of 4th Alvey Vision Conference*, pp. 147–151, 1988.

[17] J. Dittmann, T. Fiebig, and R. Steinmetz, “New approach for transformation-invariant image and video watermarking in the spatial domain: self-spanning patterns (ssp),” *Proceedings of SPIE: Security and Watermarking of Multimedia Contents II*, vol. 3971, pp. 176–185, 2000.

[18] O. Aichholzer, F. Hurtado, and M. Noy, “A lower bound on the number of triangulations of planar point sets,” *Computational Geometry: Theory and Applications*, vol. 29, no. 2, pp. 135–145, 2004.

[19] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, 2nd ed. Springer-Verlag, 2000.

[20] S. Hu, “Document image watermarking algorithm based on neighborhood pixel ratio,” *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005.

[21] J. Smith and B. Comiskey, “Modulation and information hiding in images,” *Proceedings of Information Hiding Workshop*, 1996.

[22] M. Celik, G. Sharma, and A. Tekalp, “Collusion-resilient fingerprinting by random pre-warping,” *IEEE Signal Processing Letters*, vol. 11, no. 10, pp. 831 – 835, 2004.

[23] M. Wu, W. Trappe, Z. Wang, and K. Liu, “Collusion-resistant fingerprinting for multimedia,” *Signal Processing Magazine, IEEE*, vol. 21, no. 2, pp. 15 – 27, 2004.

[24] D. Boneh and J. Show, “Collusion-secure fingerprinting for digital data,” *IEEE Transactions on Information Theory*, vol. 44, no. 5, pp. 1897–1905, 1998.

[25] J. Dittmann, A. Behr, M. Stabenau, P. Schmitt, J. Schwenk, and J. Ueberberg, “Combining digital watermarks and collision secure fingerprints for digital images,” *Proceedings of SPIE: Security and Watermarking of Multimedia Contents*, pp. 171–182, 1999.

[26] J. Su, J. Eggers, and B. Girod, “Capacity of digital watermarks subjected to an optimal collusion attack,” *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2000.

[27] J. Domingo-Ferrer and J. Herrera-Joancomarti, "Simple collusion-secure fingerprinting schemes for images," *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC)*, 2000.

[28] J. Dittmann, P. Schmitt, E. Saar, J. Schwenk, and J. Ueberberg, "Combining digital watermarks and collusion secure fingerprints for digital images," *Journal of Electronic Imaging*, vol. 9, 2000.

[29] Z. Wang, M. Wu, W. Trappe, and K. Liu, "Anti-collusion of group-oriented fingerprinting," *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2003.

[30] F. Petitcolas, R. Anderson, and M. Kuhn, "Attacks on copyright marking systems," *Proceedings of the Second Information Hiding Workshop, LNCS vol. 1525, Springer-Verlag, New York*, pp. 219–239, 1998.

[31] F. Petitcolas, "Watermarking schemes evaluation," *IEEE Signal Processing*, vol. 17, no. 5, pp. 58–64, 2000.

[32] M. Cutts, "An introduction to the gimp," *http://www.acm.org/crossroads/xrds3-4/gimp.html*, 1997.

[33] M. Kutter, S. Bhattacharjee, and T. Ebrahimi, "Towards second generation watermarking schemes," *Proceedings of International Conference on Image Processing (ICIP)*, vol. 1, pp. 320 – 323, 1999.