EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Local distortion resistant image watermarking relying on salient feature extraction

Athanasios Nikolaidis

Abstract

The purpose of this article is to present a novel method for region based image watermarking that can tolerate local image distortions to a substantially greater extent than existing methods. The first stage of the method relies on computing a normalized version of the original image using image moments. The next step is to extract a set of feature points that will act as centers of the watermark embedding areas. Four different existing feature extraction techniques are tested: Radial Symmetry Transform (RST), scale-invariant feature transform (SIFT), speeded up robust features (SURF) and features from accelerated segment test (FAST). Instead of embedding the watermark in the DCT domain of the normalized image, we follow the equivalent procedure of first performing the inverse DCT of the original watermark, inversely normalizing it and finally embedding it in the original image. This is done in order to minimize image distortion imposed by inversely normalizing the normalized image to obtain the original. The detection process consists of normalizing the input image and extracting the feature points of the normalized image, after which a correlation detector is employed to detect the possibly inserted watermark in the normalized image. Experimental results demonstrate the relative performance of the four different feature extraction techniques under both geometrical and signal processing operations, as well as the overall superiority of the method against two state-of-the-art techniques that are quite robust as far as local image distortions are concerned.

Keywords: digital image watermarking, local image distortions, image moments, radial symmetry transform, discrete cosine transform, feature extraction, SIFT, SURF, FAST

1 Introduction

During the last two decades there has been a great increase in the amount of multimedia information exchanged through the Internet. This resulted in the need for an efficient way to protect copyright on this information. The most sophisticated method to accomplish this in present years is digital watermarking [1-3]. It is interesting to note that it has since been also used in the context of other applications such as integrity checking [4,5], broadcast monitoring [6,7] and fingerprinting [8,9]. When referring to the design of a watermarking algorithm for copyright protection of digital images, there are certain requirements that we would like it to meet [10]:

- *Robustness*: The watermark should be resistant against intentional or unintentional attacks. That means, it should not be easy to render it undetectable or to remove it.
- *Imperceptibility:* The watermark should be invisible. Specifically, it should not affect the overall quality of the original image.
- *Security:* There should exist a large set of different possible keys producing independent watermarks. One should not be able to decide which the embedding key was.
- *Capacity:* It should be possible to embed and, subsequently, detect multiple watermarks in the same image.
- *Payload*: The number of watermark bits that could be embedded should be high.

As one can imagine, it is difficult to fulfill all requirements to the greatest extent simultaneously. A tradeoff

Correspondence: nikolaid@teiser.gr
Department of Informatics and Communications, Technological Educational Institute of Serres, Serres, Greece

should rather be established. In our article, we choose to focus on the robustness requirement having in mind that it is difficult to ensure a high degree of robustness without increasing watermark energy to a level that renders the watermark visible. On the other hand, if watermark energy remains low to ensure invisibility, it is unlikely that the watermark will survive any possible attack. The proposed technique, as will be shown, achieves to balance between these two requirements. Payload is kept at a moderate level, although rather small embedding areas are used for our multibit method and the adapted watermark pattern is duplicated across all of them. Finally, security and capacity remain high.

Possible watermark attacks can be categorized as follows:

- *Geometrical attacks:* these include scaling, shearing, rotation, combinations of them and local distortions such as Stirmark attack or line removal.
- *Signal processing attacks:* examples are lowpass filtering, lossy compression and noise addition.

Most of the proposed methods to date focus on either of these attack categories. The choice of embedding domain and the watermark's shape are two factors that determine which attack category the watermark is more resistant to. In general, watermarks embedded in the spatial domain can be designed in such a way that synchronization after geometric attacks can be achieved, whereas embedding in a transform domain usually provides greater robustness against filtering and compression. Additionally, watermarks having a certain symmetry (usually circular, as in [11,12]) are employed to cope with geometrical attacks. Certain methods proposed in the recent years tend to be robust against both attack categories. In [13], a scheme is described that involves image segmentation, Gaussian scale model and moment normalization of selected circular regions. The problem encountered in this method is that the inverse normalization of the embedding regions may result in boundary artifacts. Apart from that, the homogeneity criterion of the employed segmentation method cannot provide a stable representation of the image after watermark embedding and/or some attack. In [14], a drawback is the fact that the strongest corner points detected are not necessarily the mostly repeated, i.e., corner strength does not change proportionally for all points after some attack. Another problem is the increased complexity due to both circular convolution needed to ensure rotational invariance and local search needed to overcome instability of feature point position and scale. The methods proposed in [15,16] also suffer from quantization error due to inverse normalization of the embedding disks although some remedies are proposed in [15] to overcome this. These remedies, however, may affect detector performance. Besides, in [15] the number of correctly detected feature points after watermarking and possible attacks affects the detection threshold used to decide on the existence of the watermark. The watermark embedded using the technique described in [17] cannot withstand shearing attacks and, consequently, any affine geometrical attack involving shearing. That is because of the fact that the watermark is only rotationally invariant due to its structure of homocentric cirques and scaling invariant due to prior scale normalization of the whole image. Finally, in [18], a method is proposed that utilizes the scale-invariant feature transform (SIFT) to extract circular patches that are scale and translation invariant, and the prototype rectangular watermark is subsequently inversely polar-mapped prior to embedding. However, a computational overhead is introduced, again, due to circular convolution needed during detection to compensate for image rotation and, eventually, decide on the existence of the watermark.

In the following sections we describe a watermarking technique that deals successfully with all of the problems stated above and, additionally, provides substantially greater robustness than existing methods against local distortions, while keeping robustness against other usual attacks at an acceptable level. In Section 2, the initial stage of preprocessing which precedes both watermark embedding and detection is first described. In Section 3, the main watermarking procedure is explained and Section 4 presents examples of experimental results that prove the efficiency of the technique. Finally, conclusions about this study are drawn in Section 5.

2 Image preprocessing

Both watermark embedding and detection procedures require that a proper preprocessing of the original image has taken place, so that the watermark embedding or detection areas can be located. Section 2.1 describes the first preprocessing step where the original image is transformed geometrically to a standard form. Section 2.2 briefly overviews the four different feature extraction methods that will alternatively act upon the normalized image to produce the reference points both for watermark embedding and detection.

2.1 Image normalization

The first step prior to watermark embedding and detection is image normalization. This serves to provide the next step of feature extraction with a standard form of the original image, in which to search for strong feature points. The difference from other methods in the literature is that they employ image normalization in circular patches that have already been extracted from the original image. The problem, as stated in Section 1, is that the

normalized and afterwards watermarked patches have to be inversely normalized and overlayed on the original image, leading to interpolation errors and, thus, visible artifacts. In the current article, we implement the image normalization method proposed in [19]. Here we should point out that the method described in [19] is the first step of a watermarking technique which, however, affects the whole of the image. Our aim in the present article is to provide a technique that only affects the image regionally, since we wish to cope with local image distortions. If we let $I(x, y)$ be the original image, then the normalized image is $g(x, y) = I(x_\alpha, y_\alpha)$, where

$$\begin{pmatrix} x_\alpha \\ y_\alpha \\ 1 \end{pmatrix} = \mathbf{S} \cdot \mathbf{Y} \cdot \mathbf{X} \cdot \mathbf{T} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{1}$$

and $\mathbf{T} = \begin{pmatrix} 1 & 0 & -d1 \\ 0 & 1 & -d2 \\ 0 & 0 & 1 \end{pmatrix}$ is a translation matrix, $\mathbf{X} = \begin{pmatrix} 1 & \beta & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ is a x-shearing matrix, $\mathbf{Y} = \begin{pmatrix} 1 & 0 & 0 \\ \gamma & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ is a y-shearing matrix, and $\mathbf{S} = \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \delta & 0 \\ 0 & 0 & 1 \end{pmatrix}$ is a scaling matrix.

The values of the parameters d_1, d_2 are calculated as

$$d_1 = \frac{m_{10}}{m_{00}}, d_2 = \frac{m_{01}}{m_{00}}$$

where m_{10}, m_{01}, m_{00} are geometric moments of the original image $I(x, y)$

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p y^q I(x, y) \tag{3}$$

If we let $I_T(x, y)$ be the image after translation normalization, the value of the parameter β is calculated as a root of

$$\mu_{30}^{(T)} + 3\beta\mu_{21}^{(T)} + 3\beta^2\mu_{12}^{(T)} + \beta^3\mu_{03}^{(T)} = 0 \tag{4}$$

where $\mu_{pq}^{(T)}$ are the central moments of $I_T(x, y)$

$$\mu_{pq}^{(T)} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x - \bar{x})^p (y - \bar{y})^q I_T(x, y) \tag{5}$$

In case of a singe real root and two complex conjugate roots, the value of β is chosen as the real one. In case of three real roots, the value is chosen as the median. The value of γ is calculated as

$$\gamma = -\frac{\mu_{11}^{(XT)}}{\mu_{20}^{(XT)}} \tag{6}$$

where $\mu_{pq}^{(XT)}$ are the central moments of $I_{XT}(x, y)$ which is the image $I_T(x, y)$ after x-shearing normalization. Finally, the values of α and δ are derived given that $I_{Y\,XT}(x, y)$ (the image $I_{XT}(x, y)$ after y-shearing normalization) is resized to a specific size (e.g., 512 × 512 in our experiments) to provide the final normalized image $I_{SY\,XT}(x, y)$. The signs of these parameter values are determined by the constraint that both $\mu_{50}^{(SY\,XT)}$ and $\mu_{05}^{(SY\,XT)}$ are positive. Examples of the original "Lena" and "Lake" images and respective normalized images using the above described method are shown in Figure 1.

This normalized representation of the original image is the input for the next step of preprocessing that is necessary for both watermark embedding and detection.

2.2 Feature extraction

The second step of the preprocessing stage is the feature extraction step. A great variety of feature extraction methods has been proposed in the literature. Lately, there is a tendency of using the so-called *scale-space* methods such as SIFT [20] for watermarking purposes [18,21-23]. In our study, we employed this as well as other feature detectors proposed in the literature, but not in the context of image watermarking, during the past few years. These detectors are, more specifically, the radial symmetry transform (RST) introduced in [24], the speeded up robust features (SURF) [25,26] and the features from accelerated segment test (FAST) [27,28]. As we will show in the experimental results section, all of them perform adequately well for our application, although their relative performance varies.

2.2.1 Radial symmetry transform

To compute the RST first we have to construct two images, the *magnitude projection image* M_n and the *orientation projection image* O_n of the normalized image at every radius n that we have selected. These images are initialized to zero and are subsequently updated at each point depending on how the point is affected by the gradient vector at a point a distance n away. Let $\mathbf{p} = (x, y)$ be a point and $g(\mathbf{p})$ the gradient vector at that point, determined by applying the 3 × 3 Sobel operator at the respective point of the normalized image. The coordinates of the so-called *positively-affected* pixel are

$$\mathbf{p}_{+ve}(\mathbf{p}) = \mathbf{p} + \text{round}\left(\frac{g(\mathbf{p})}{\|g(\mathbf{p})\|} n\right), \tag{7}$$

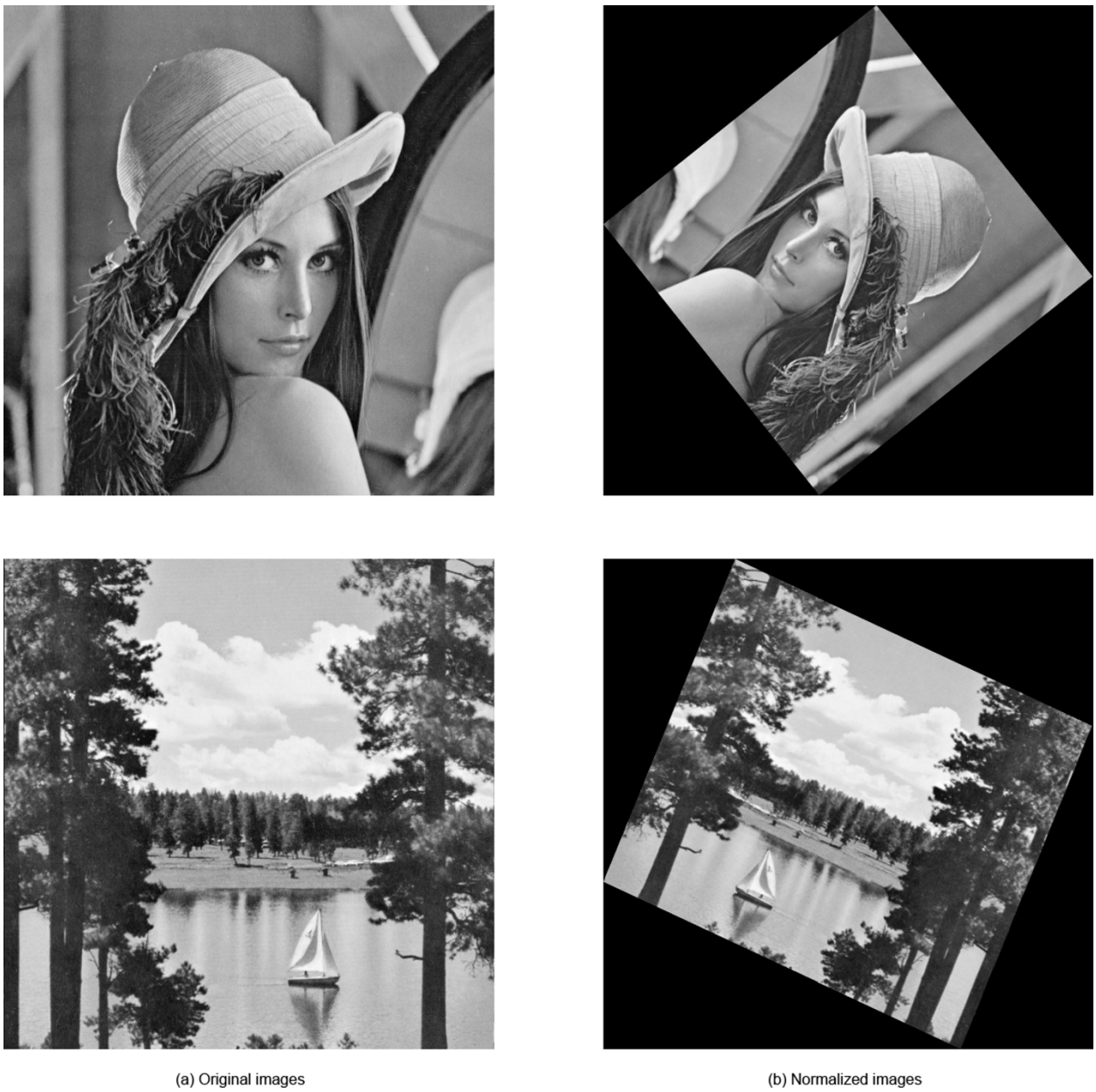


Figure 1 Results for image normalization.

and those of the *negatively-affected* pixel are

$$\mathbf{p}_{-ve}(\mathbf{p}) = \mathbf{p} - \text{round}\left(\frac{g(\mathbf{p})}{\|g(\mathbf{p})\|}n\right), \tag{8}$$

The pixel values of the magnitude projection and orientation projection images are updated as follows

$$M_n(\mathbf{p}_{+ve}(\mathbf{p})) = M_n(\mathbf{p}_{+ve}(\mathbf{p})) + \|g(\mathbf{p})\|, \tag{9}$$

$$M_n(\mathbf{p}_{-ve}(\mathbf{p})) = M_n(\mathbf{p}_{-ve}(\mathbf{p})) - \|g(\mathbf{p})\|, \tag{10}$$

$$O_n(\mathbf{p}_{+ve}(\mathbf{p})) = O_n(\mathbf{p}_{+ve}(\mathbf{p})) + 1, \tag{11}$$

$$O_n(\mathbf{p}_{-ve}(\mathbf{p})) = O_n(\mathbf{p}_{-ve}(\mathbf{p})) - 1. \tag{12}$$

Next, we have to define

$$\tilde{O}_n(\mathbf{p}) = \begin{cases} O_n(\mathbf{p}) & \text{if } O_n(\mathbf{p}) < k_n \\ k_n & \text{otherwise.} \end{cases} \tag{13}$$

where k_n is a scaling factor to normalize M_n and O_n across different radii. Once $\tilde{O}_n$ is defined, we compute

$$F_n(\mathbf{p}) = \frac{M_n(\mathbf{p})}{k_n}\left(\frac{|\tilde{O}_n(\mathbf{p})|}{k_n}\right)^{\alpha}, \tag{14}$$

where α is the *radial strictness parameter*. The larger the value of α, the stricter the required radial symmetry. Finally, F_n is convolved with a 2D Gaussian filter A_n to produce the radial symmetry contribution at radius n

$$S_n = F_n * A_n \tag{15}$$

The overall RST (symmetry map) is calculated by simply averaging radial symmetry contributions for all of the radii considered

$$S = \frac{1}{|N|}\sum_{n\in N} S_n \tag{16}$$

where N is the set of radii. A non-maximum suppression and thresholding algorithm [29] is applied to the symmetry map S to localize the strongly symmetric points of the normalized image. An example for the images of Figure 1 is depicted in Figure 2 for $N = \{1, 3, 5\}$ and $\alpha = 1$. The value of the radius for non-maximum suppression was chosen to be 3 and that of the threshold to be 5.

2.2.2 Scale-invariant feature transform

The main idea of this detector is to search for candidate stable feature points across a series of image scales. First, the so-called *scale space* of the normalized image is constructed by convolving the image $I(x, y)$ with a variable-scale Gaussian $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2}e^{-(x^2+y^2)/2\sigma^2}$

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{17}$$

The potentially stable feature points are detected as local extrema of the function $D(x, y, \sigma)$ constructed as follows

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{18}$$

that is, a convolution of the image with a difference of Gaussians. k is a factor that determines the difference between consecutive scales. An *octave* of scale space is a series of $D(x, y, \sigma)$ functions spanning a doubling of σ. Each octave is divided in s intervals and, thus, $k = 2^{1/s}$. For each new octave, the Gaussian image produced with the doubled value of σ at the previous octave is first downsampled by a factor of 2 at each dimension. The local minima and maxima are found by 3D search in the 8 neighbors of the current scale and the respective 9 neighbors in each of the previous and the next scale.

To correctly localize feature points, candidate points are fitted to the nearby data by interpolation. The Taylor expansion of the function $D(x, y, \sigma)$ is given by

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x} \tag{19}$$

where D and its derivatives are calculated at the candidate feature point and $\mathbf{x} = (x, y, \sigma)^T$ is the offset from this point. The location of the extremum $\hat{\mathbf{x}}$ is found by taking the derivative of this expansion and setting it to zero, giving

$$\hat{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2}\frac{\partial D}{\partial \mathbf{x}} \tag{20}$$

If the offset $\hat{\mathbf{x}}$ is larger than 0.5 in any dimension, then the extremum should be closer to another candidate feature point. If so, the interpolation is again performed around a different point. Otherwise the offset is added to the candidate point to produce the interpolated estimate of the extremum.

To discard feature points of low contrast, the value of the second-order Taylor expansion is computed at the offset $\hat{\mathbf{x}}$. If this value is less than 0.03 then the candidate point is discarded. Otherwise it is kept, and its final location and scale are, respectively, $\mathbf{y} + \hat{\mathbf{x}}$ and σ, where $\mathbf{y}$ is the original location of the candidate point at scale σ.

Another action that should be taken is to eliminate feature points with strong edge response. To do so, we first have to compute the second-order Hessian matrix $\mathbf{H}$

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{21}$$

whose eigenvalues are proportional to the principal curvatures of D. If we let α be the larger eigenvalue and β the smaller one, then it can be shown that

$$R = \frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} = \frac{(r+1)^2}{r} \tag{22}$$

where $r = \alpha/\beta$, $Tr(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta$ is the trace of $\mathbf{H}$ and $Det(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta$ is the determinant of $\mathbf{H}$. If the ratio R for a certain candidate feature point is larger than $(r_{th} + 1)^2/r_{th}$, then the feature point is rejected. The method sets the threshold eigenvalue ratio to $r_{th} = 10$.

In our experiments the values of the various parameters involved in this method were chosen in accordance with [20]. Only the strength threshold for local maxima of the scale space was chosen to be equal to 0.05 to reduce the number of produced feature points. Examples of feature points extracted from the normalized versions of “Lena” and “Lake” are shown in Figure 3.

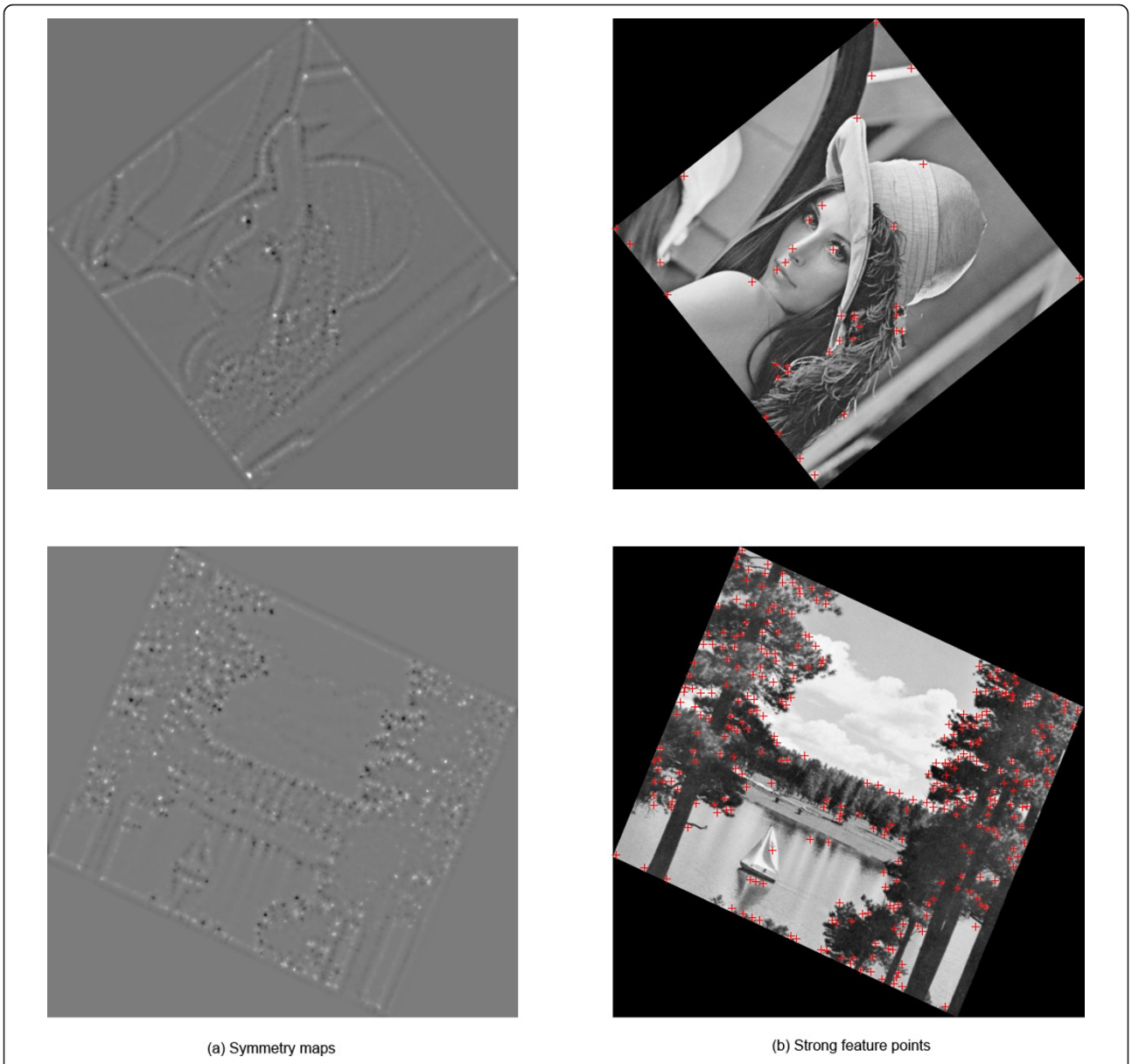


Figure 2 Symmetry maps and strong feature points of normalized images.

2.2.3 Speeded up robust features

This method was introduced as an alternative to SIFT focusing on computational cost reduction. A fast way of computing the Hessian matrix using integral images is proposed. This approach approximates the second order Gaussian derivatives by box filters. These, in turn, are used to compute the approximate determinant of the Hessian matrix. Instead of subsampling the filtered image of a previous layer, the scale space is constructed by increasing the filter size. For each new octave, the filter size increase per layer is doubled, and so is the sampling interval for the extracted feature points.

In the experiments that we conducted, the number of octaves that were analyzed was 5, the initial sampling interval was 2 and the Hessian response threshold was chosen to be 0.004. The feature points extracted from the normalized versions of "Lena" and "Lake" are presented in Figure 4.

2.2.4 Features from accelerated segment test

This feature detector should be more precisely called a corner detector. To test if a certain pixel p is a corner, 16 pixels lying on a circle centered at this pixel (specifically, a Bresenham circle of radius 3) are tested for similarity of intensity to the center pixel. If N contiguous

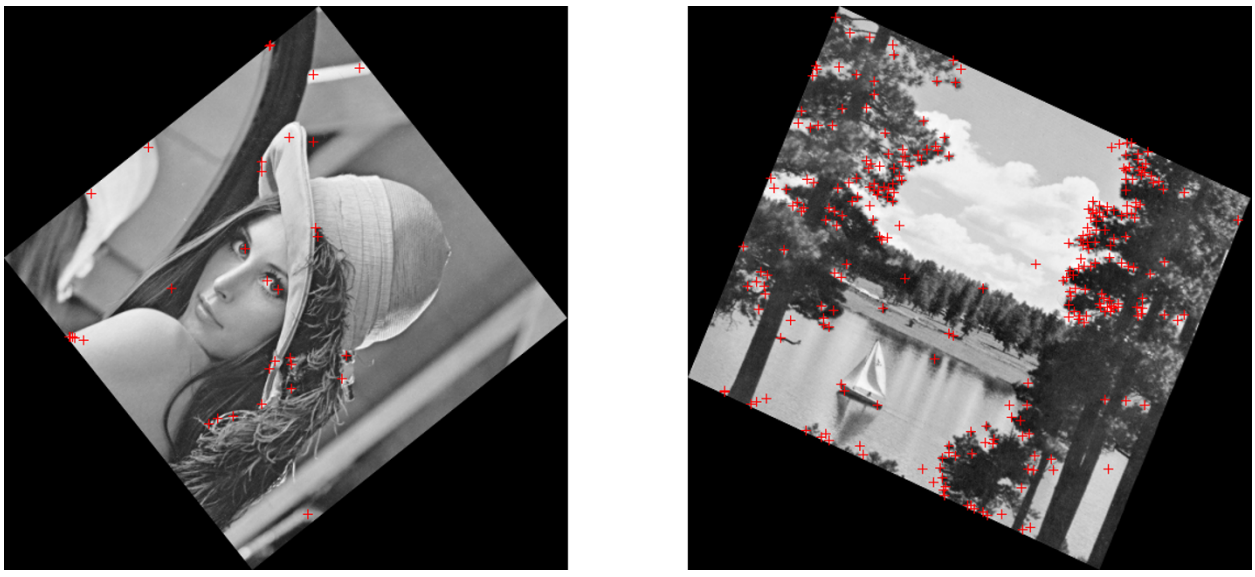


Figure 3 Feature points extracted using SIFT.

pixels lying on this circle are all brighter than the center pixel by a quantity T (that is $I_{p\to x} \geq I_p+T$, $x \in \{1 \ldots 16\}$) or darker than it by the same quantity (that is $I_{p\to x} \leq I_p - T$, $x \in \{1 \ldots 16\}$), then the center pixel is considered a corner. A non-maximum suppression step follows to reduce the number of corner points. Since there is no score function on which to apply the suppression, we define one as [28]

$$V = max\left(\sum_{x\in S_{\text{bright}}} |I_{p\to x} - I_p| - T, \sum_{x\in S_{\text{dark}}} |I_p - I_{p\to x}| - T\right) \quad (23)$$

where

$$\begin{aligned} S_{\text{bright}} &= \{x | I_{p\to x} \geq I_p + T\} \\ S_{\text{dark}} &= \{x | I_{p\to x} \leq I_p - T\} \end{aligned} \quad (24)$$

After suppression only the candidates having score value greater than all their 8 neighbors are preserved. The parameter values used in our experiments where $N = 12$ and $T = 60$. For the "Lena" and "Lake" images, the feature points extracted from their normalized versions are shown in Figure 5.

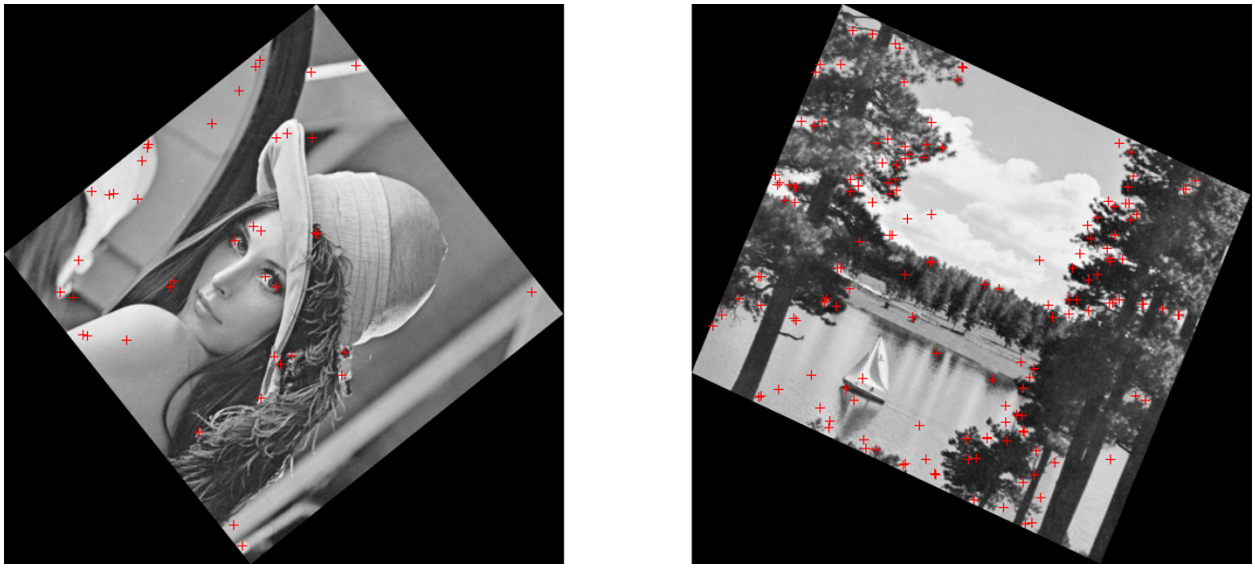


Figure 4 Feature points extracted using SURF.

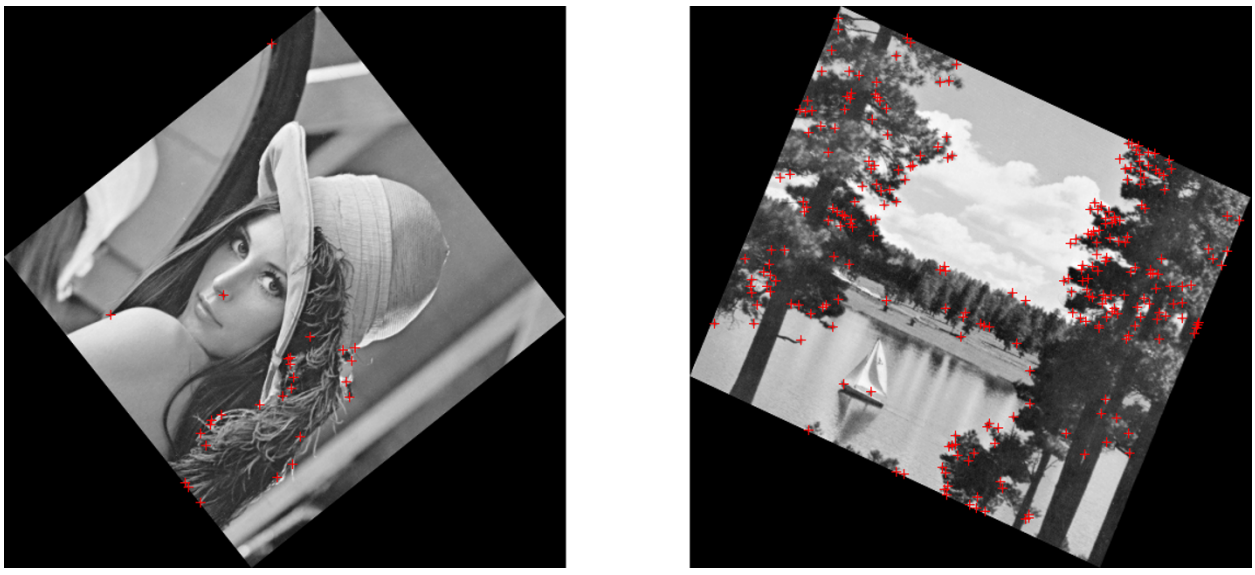


Figure 5 Feature points extracted using FAST.

3 Watermarking scheme

The preprocessing stage described in the previous section is, as already stated, common for both watermark embedding and detection procedures. The extracted feature points are to be used as centers of the areas where the watermark is to be embedded.

The watermark pattern is initially constructed in the DCT domain as a rectangular patch of size that is related to the size of the normalized image (e.g., 64 × 64 for a normalized image of size 512 × 512, as in our examples). Other methods employing DCT in the field of image watermarking have been proposed in the past as well [30]. If we let $\mathbf{b}_i$, $i = 1, \ldots, N$ be binary sequences of length K (which is the number of DCT coefficients that are going to be modulated) created by thresholding pseudorandom values taken from the standard normal distribution (i.e., $\mathcal{N}(0, 1)$), where N is the length of the multibit watermark message, and m_i is the ith bit of the message, then the middle zone of K DCT coefficients is modulated as follows:

$$\mathbf{C} = \sum_{i=1}^{N} (2m_i - 1)\mathbf{b}_i \tag{25}$$

The position of the middle zone of DCT coefficients is chosen so as to render the watermark both robust to attacks that affect high frequencies (such as JPEG compression or lowpass filtering) and invisible (by preserving low frequency content). The rest of the DCT coefficients are set to zero. The final watermark pattern is produced by inverse zig-zag scanning of the zero-padded $\mathbf{C}$ sequence. An example of such a watermark (of size 64 × 64) and its spatial counterpart (its inverse DCT) is depicted in Figure 6. The range of non-zero coefficients is chosen to be [407, 3316] in the zig-zag order, which means that $K = 2910$. We can notice the non-white properties of the watermark pattern in the spatial domain representation.

3.1 Watermark embedding

The original aim is to insert the watermark in the DCT transform domain - other domains such as the space/spatial-frequency domain [31] could alternatively be employed - of the normalized image or, equivalently, insert the inverse DCT of the watermark in the spatial domain of the normalized image. However, by doing so, we would afterwards have to inversely normalize the watermarked normalized image to obtain the watermarked original image so that the watermark embedding process would be complete. This, as pointed out in Section 1, would impose interpolation errors, resulting in a version of the image that would be visibly corrupt compared to the original, even in areas that would not be normally affected by watermark embedding. To avoid this image degradation we choose to embed the inversely normalized version of the inverse DCT of the original watermark in the original image. Additionally, the watermark is to be embedded in all areas corresponding to the extracted feature points of the normalized image in a similar fashion as in [32]. This is done to increase watermark robustness as it is possible that not all originally detected feature points will also be detected after some attack. The overall embedding procedure is depicted in Figure 7.

(a) DCT domain watermark (b) Spatial domain watermark

Figure 6 Original DCT-formed watermark and its spatial domain counterpart.

More formally, for each embedding area $g_i(x, y)$, $i = 1 \ldots M$ (where M is the number of feature points) of the normalized image we additively embed the DCT-domain watermark as follows

$$DCT(g_i^w(x,y)) = DCT(g_i(x,y)) + \alpha \cdot W(u,v) \tag{26}$$

where $W(u, v)$ is the original DCT-domain watermark and α is the embedding strength. Given that the DCT is an orthogonal transform, Equation (26) can be rewritten as

$$\begin{aligned} g_i^w(x,y) &= g_i(x,y) + IDCT(\alpha \cdot W(u,v)) \\ &= g_i(x,y) + \alpha \cdot w(x,y) \end{aligned} \tag{27}$$

where $w(x, y)$ is the inverse DCT of $W(u, v)$. If we followed directly this procedure for watermark embedding we would, eventually, have to inversely normalize the watermarked normalized image $g^w(x, y)$ to produce the watermarked version $f^w(x, y)$ of the original image:

$$\begin{pmatrix} x_b \\ y_b \\ 1 \end{pmatrix} = \mathbf{T}^{-1} \cdot \mathbf{X}^{-1} \cdot \mathbf{Y}^{-1} \cdot \mathbf{S}^{-1} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{28}$$

where $f^w(x, y) = g^w(x_b, y_b)$. However, as aforementioned, the image would thus be visibly damaged. Instead of performing embedding according to Equation (27), we choose to embed the watermark directly in the original image. To do so, we have to inversely normalize the upright rectangular watermark pattern and embed it in the original image, centered at the points that correspond to the feature points extracted from the normalized image:

$$f_i^w(x,y) = f_i(x,y) + \alpha \cdot w_o(x,y) \tag{29}$$

where $w_o(x, y) = w(x_b, y_b)$ according to Equation (28), $f_i(x, y)$ with $i = 1 \ldots M$ are the areas of the original image where the watermark is to be embedded and $f_i^w(x,y)$ are the respective watermarked areas. An example of a watermarked version of the image "Lake" of PSNR = 24.69 dB using RST and its amplified difference from the original is given in Figure 8. We can notice that some embedding areas may overlap because of the proximity of the corresponding feature points. We prefer to use all feature points as embedding area centers instead of applying some kind of criterion to select some of them. That is because we cannot be certain about the repeatability of feature points (that is, the probability that a specific point will be extracted in any altered version of the image). Since the watermark is embedded around all extracted points, it is also going to be detected around all feature points extracted during the detection stage, as it will be described in the following section. Thus, to cover the case of overlapping areas, it would be more appropriate to describe embedding in an iterative manner

$$f_i^w(x,y) = f_{i-1}^w(x,y) + \alpha \cdot w_i(x,y) \tag{30}$$

where $i = 1, \ldots, M$, $w_i(x, y)$ is the image with same size as $f(x, y)$ and non-zero only in the ith embedding area (where w_o is located), and $f_0^w(x,y) = f(x,y)$.

An evident problem that may arise because of watermark area overlapping is that the watermark might

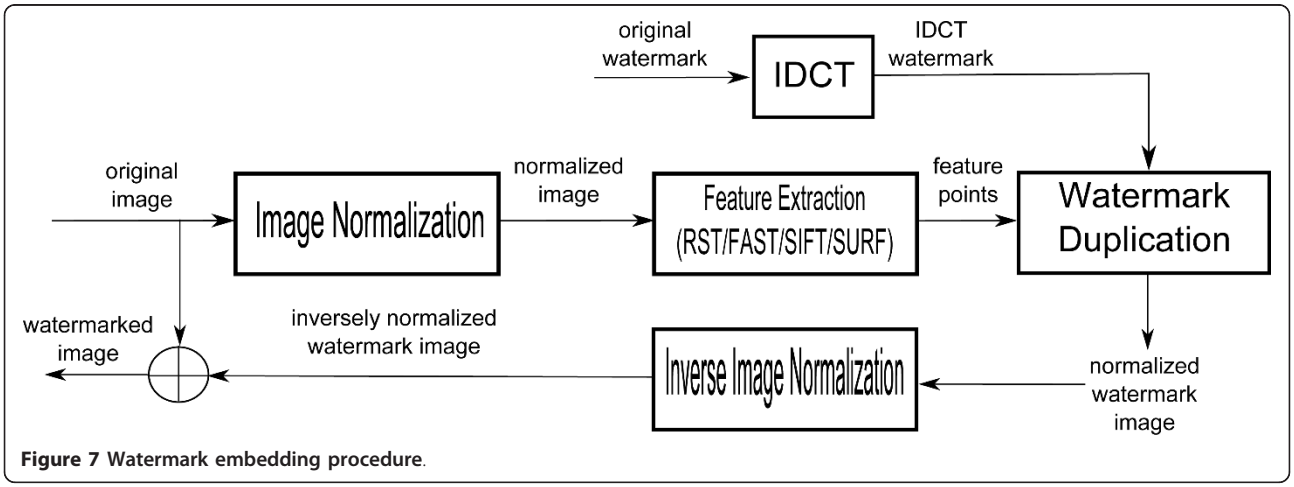
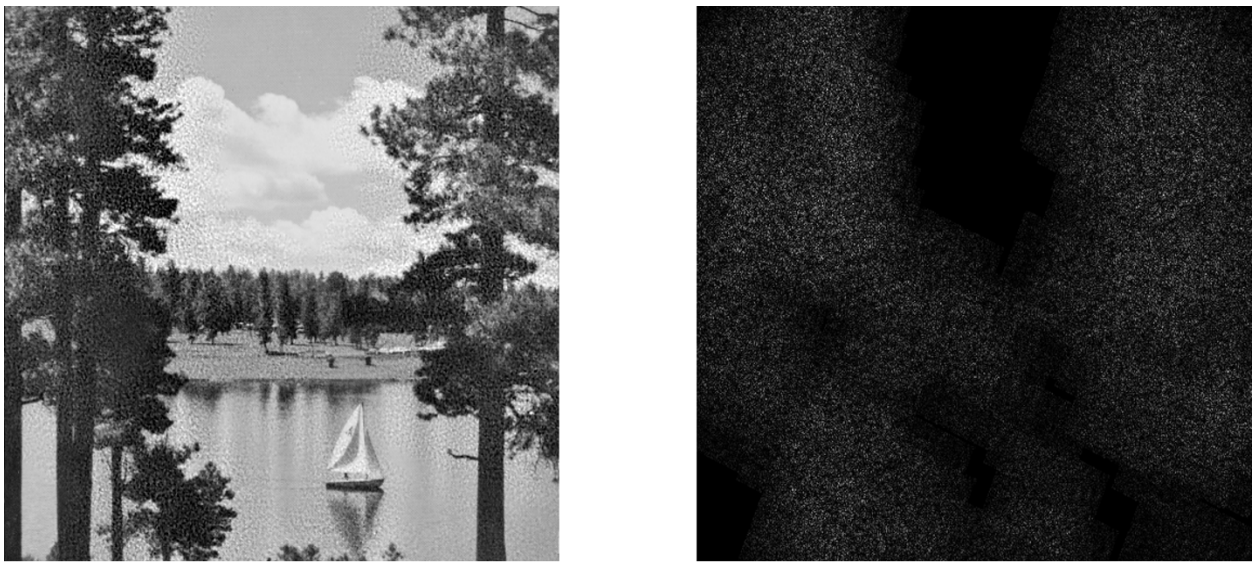


Figure 7 Watermark embedding procedure.



(a) Watermarked image (b) Amplified difference

Figure 8 Watermarked image without visibility rule and its amplified difference from the original.

become visible, as one can see in Figure 8. To overcome this, we modify Equation (30) in the following way

$$f_i^w(x,y) = f_{i-1}^w(x,y) + \alpha \cdot \frac{1}{r(x,y)} \cdot w_i(x,y) \tag{31}$$

where $r(x, y)$ is the number of watermarked areas overlapping at point (x, y). If no watermarking has occurred at that point, then $r(x, y) = 1$. A non-iterative version of Equation (31) is

$$f_M^w(x,y) = f(x,y) + \alpha \cdot \frac{1}{r(x,y)} \cdot \sum_{i=1}^{M} w_i(x,y) \tag{32}$$

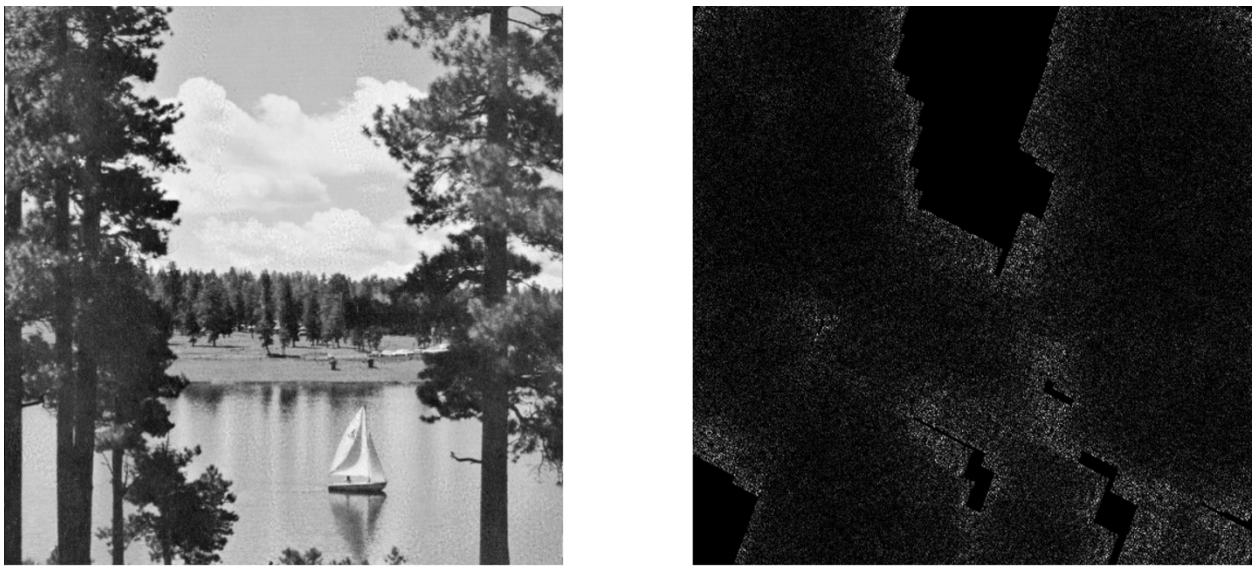
An example of applying this rule is given in Figure 9. The watermarked image now has PSNR = 40 dB and, in contrast to Figure 8, the watermark is hardly visible.

3.2 Watermark detection

To perform watermark detection, the preprocessing step is needed as for watermark embedding. This means that the watermarked and, possibly, attacked image is first geometrically normalized and feature extraction is performed in the normalized image in the same manner as in the embedding stage (using one of the methods described in Section 2.2). Figure 10 shows the result for the watermarked image of Figure 9. As one can see, a great percentage of the originally extracted feature points used for watermark embedding (see Figure 2) are still present in the normalized watermarked image. Therefore, the watermark will be detected accurately in all respective areas. Since, as pointed out in Section 3.1, no algorithm for selection of certain feature points has been established, watermark detection is going to be performed in all corresponding areas. An outline of the detection procedure is shown in Figure 11.

Detection is performed blindly, meaning that no knowledge about the original image is required. Although embedding has been performed in the original image, detection is carried out in the normalized image. This is done to avoid the overhead of inversely normalizing the watermark, since the normalized image is already available. To decide about the value of each message bit that was originally embedded in the image, we first have to extract the sequence of DCT coefficients of each region where the watermark is supposedly embedded. If $f^{w'}(x,y)$ is the image in which the watermark is to be detected, we have to obtain its normalized version $g^{w'}(x,y)$. If we let M' be the number of extracted feature points in image $g^{w'}(x,y)$, the detector output D_j for each message bit $\hat{m}_j$ is computed by linear correlation between the respective DCT band $\mathbf{G}_i^{w'}, i = 1, \ldots, M'$ and the binary sequence $\mathbf{b}_j$ created by the same key as the one used for embedding, for all M' regions. This can be formulated as

$$D_j = \sum_{i=1}^{M'} \mathrm{corr}(\mathbf{G}_i^{w'}, \mathbf{b}_j) \tag{33}$$



(a) Watermarked image (b) Amplified difference

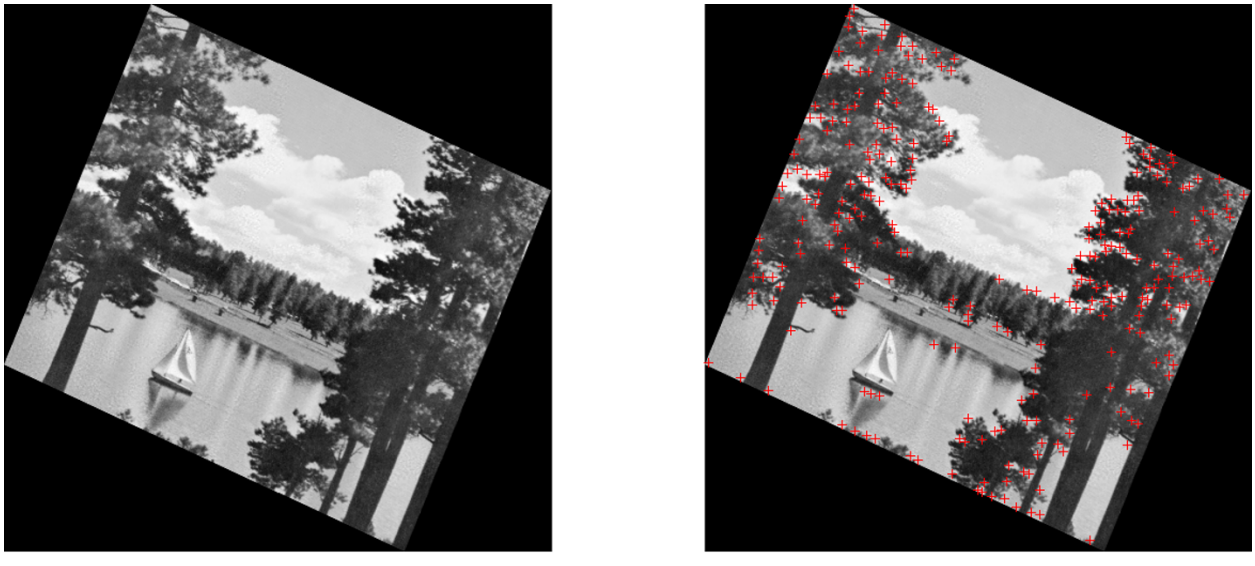
Figure 9 Watermarked image using visibility rule and its amplified difference from the original.

The value of each extracted message bit $\hat{m}_j$ can be determined by comparing the detection value D_j with zero.

$$\hat{m}_j = \begin{cases} 1, D_j > 0 \\ 0. D_j \leq 0 \end{cases} \quad (34)$$

4 Experimental results

To test the efficiency of the proposed watermarking technique against local distortions as well as other image processing attacks, we have conducted extensive watermarking experiments on ten well known images of different content, specifically "Airplane", "Boat", "House", "Peppers", "Splash", "Baboon", "Couple", "Lena", "Elaine", and "Lake". Each experiment consisted of embedding a 50 bit watermark message in each of the images and subsequently trying to extract it from the watermarked and attacked version of the image. For all techniques compared and for all images, PSNR is tuned to 40 dB. The bit error rate (BER), that is, the



(a) Normalized watermarked image (b) Corresponding feature points

Figure 10 Preprocessing for watermarked image "Lake" prior to detection.

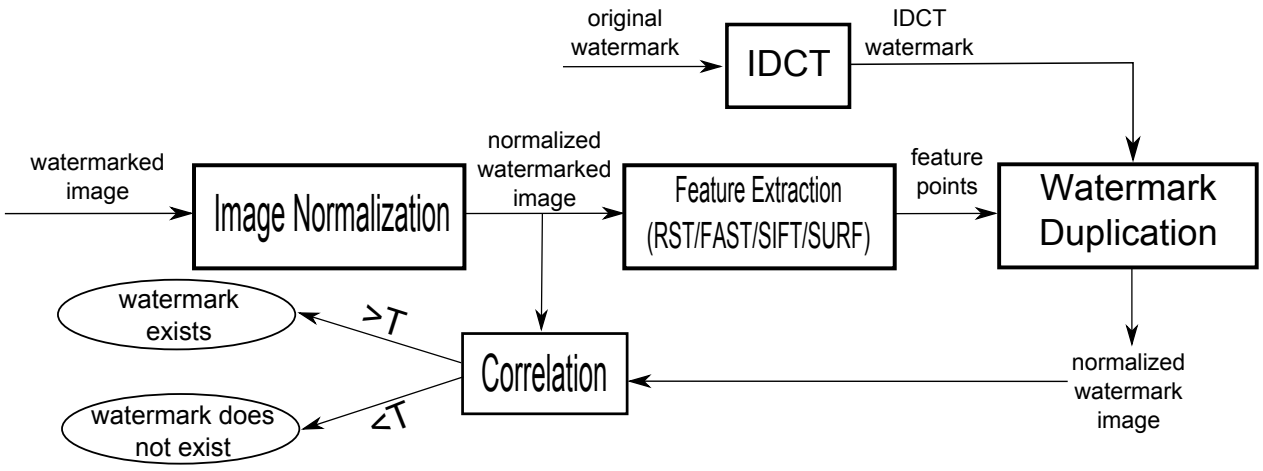


Figure 11 Watermark detection procedure.

percentage of message bits that have not been detected correctly, is finally calculated. The proposed technique was tested for all four feature detectors under concern and compared to the state-of-the-art techniques described in [19,33]. These methods were selected as two of the recent bibliography that are multibit, permit fine-tuning of PSNR and are built to resist geometric attacks. It is worth mentioning that these methods act globally, thus distorting the whole of the image. In contrast, our method affects only local regions, thus producing zero distortion in part of the image. This, in turn, results in improved imperceptibility. The parameter values for the feature detectors were those used in the examples of Section 2.2. The range of DCT coefficients used for watermarking with the technique by Dong et al. [19] was chosen to be [28681, 215478], that is 186798 coefficients. The respective range of DCT coefficients for the technique by Tian et al. [33] was [7170, 53870], that is 46701 coefficients. These ranges were chosen as equivalent to the one used in our method. In the following sections, we present results for local geometric attacks, global geometric attacks and signal processing attacks. Some of the attacks were implemented using the Checkmark benchmarking software [34].

4.1 Local geometric attacks

One classic non-geometric attack is column and line removal. In Figure 12 we can see results for this attack where the pair of values inside the parentheses denotes the number of columns and lines of the image that have been removed, and which are equidistant. We can notice that our technique performs better for all employed feature detectors. This was expected since the state-of-the-art techniques affect the image globally and cannot withstand attacks that modify image contents. The SIFT-based version of our technique demonstrated the best performance followed by RST-based and SURF-based which have similar performance and finally FAST-based which is still better than the older techniques.

The next local distortion considered was the Stirmark attack. The experiment involved varying the jitter strength parameter from 1 to 7. As one can see in Figure 13, the proposed technique is superior to the technique by Dong et al. for all versions and especially for the SIFT-based one, but the technique by Tian et al. provides better performance for all cases but one.

Another attack considered in this category was image band cropping. The idea is to crop a band of certain width around the boundaries of the image. The band width in our experiments varied from 3 to 11 pixels as one can see in Figure 14. We can notice that the state-of-the-art techniques are seriously affected even by a small amount of cropping, whereas the various versions of our technique are always more robust and slowly degrade as

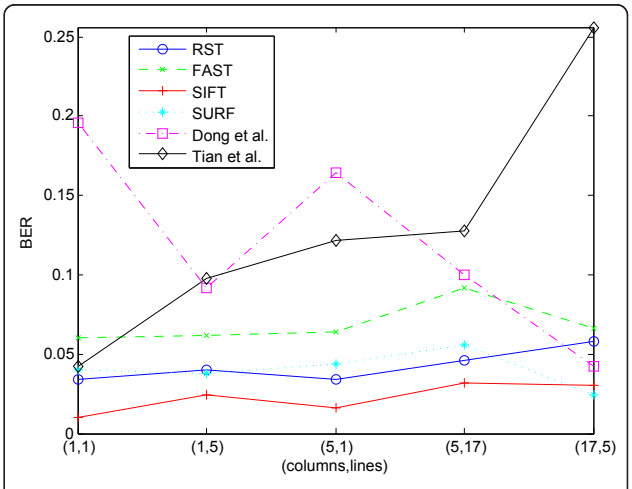


Figure 12 Column and line removal.

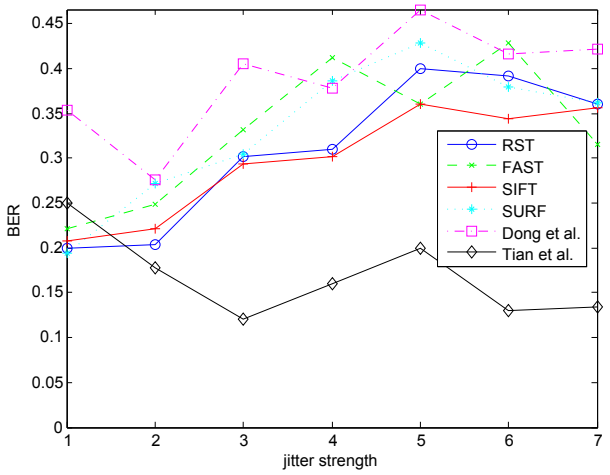


Figure 13 Stirmark attack.

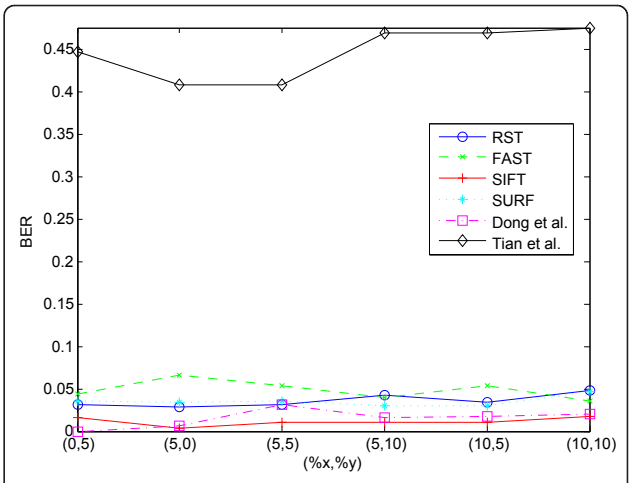


Figure 15 Shearing attack.

the band gets wider. Although all versions provide similar performance, the SIFT-based version appears to prevail. This is, again, an expected behavior since the state-of-the-art techniques are not designed to withstand attacks that severely modify the global spectral representation of the image.

4.2 Global geometric attacks

Another category of possible distortions is that of global geometric attacks. These include rotation, scaling, shearing and combinations of them (i.e., general affine transforms). The first of these attacks presented here is the shearing attack. In this experiment, the varying parameters were the shearing percentages in both x and y axes. The results shown in Figure 15 prove that the technique by Tian et al. is not resistant against such an attack, which is expected since the technique does not apply affine normalization on the original image prior to watermark embedding. Performance, however, is excellent for the rest of the methods, with the SIFT-based version providing slightly better robustness than the technique by Dong et al. which, in turn, is a little more robust than the rest of our versions.

In the case of scaling, we conducted experiments with the scaling factor taking values as shown in Figure 16. The various methods do not present great differences in performance. However, the technique by Dong et al. is the best in all cases but one. The SIFT-based version of our technique is the next in order of performance, followed by the SURF-based and the RST-based versions and the technique by Tian et al. which alternate in terms of performance for the various parameter values, and finally the FAST-based version which exhibits the lowest robustness.

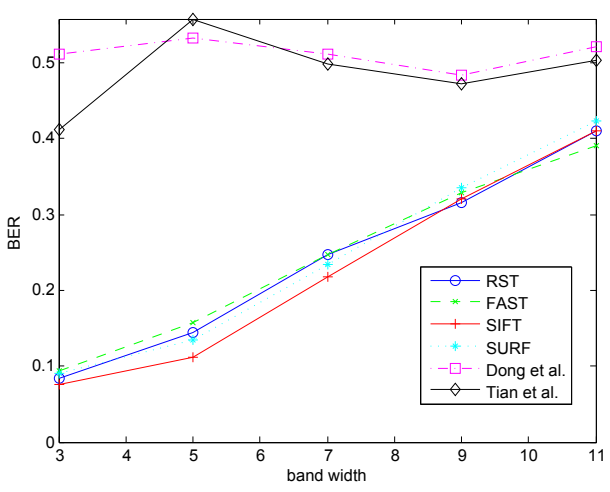


Figure 14 Band cropping.

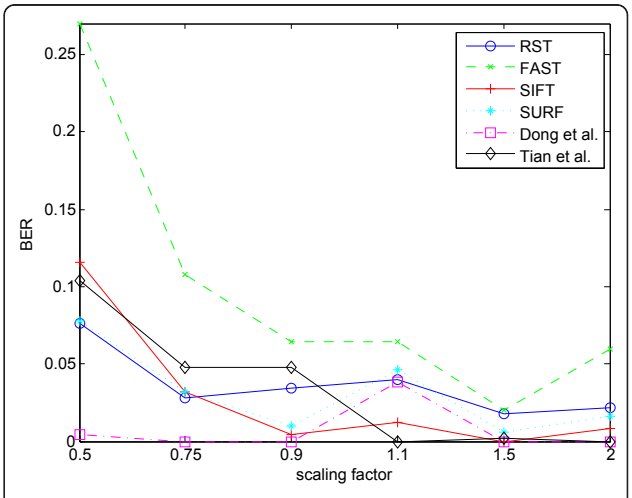


Figure 16 Scaling attack.

Rotation followed by cropping out the central region that does not contain black border pixels and finally scaling to the original size has next been tested with the varying parameter being the rotation angle, as presented in Figure 17. The technique by Tian et al. cannot withstand this attack. On the contrary, the technique by Dong et al. is superior, although the SIFT-based version of our technique is very close to it in terms of robustness, followed by SURF-based, RST-based, and FAST-based.

Another example of an attack comprising of different stages is shown in Figure 18, where successive downsampling and upsampling has been performed in the watermarked images. The pairs of values in parentheses correspond to the downsampling and upsampling factor, respectively. We can notice that all methods display similar performance with the technique by Tian et al. presenting the least varying robustness. The SIFT-based version appears to be, again, the best among all versions of our technique.

Finally, an experiment involving general affine transform was conducted, which showed that the performance of the proposed technique is comparable to that provided by the technique by Dong et al., as one can see in Figure 19. All techniques but the one by Tian et al. survive this type of attack. The varying parameters, in this case, were the affine transform matrix coefficients, considering the form $\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}$. As in the aforementioned experiments, the SIFT-based version demonstrated the best results among the four versions of our method, followed by SURF-based, RST-based, and FAST-based.

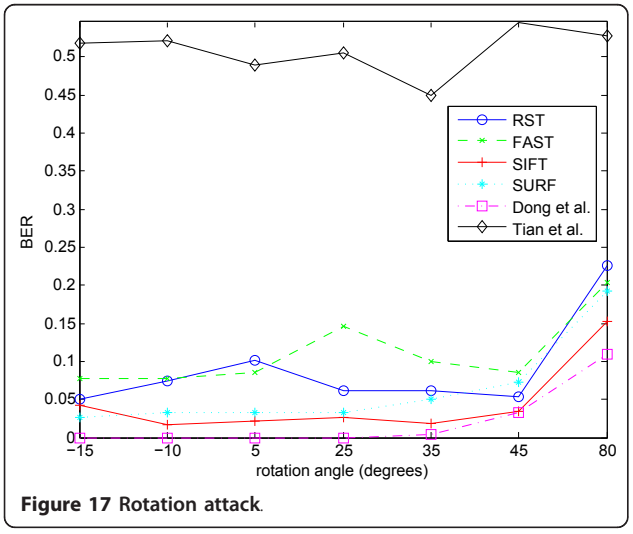


Figure 17 Rotation attack.

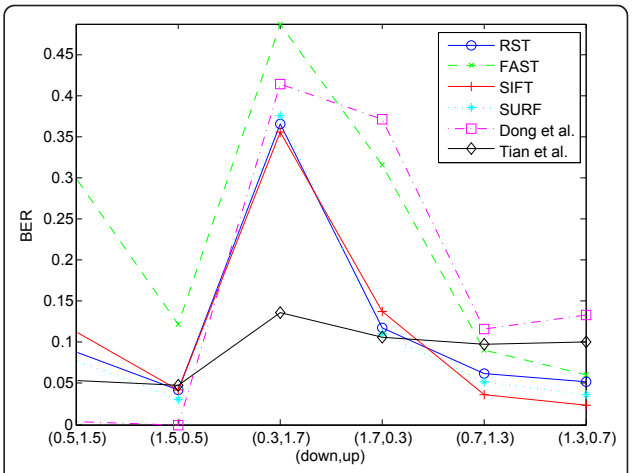


Figure 18 Downsampling followed by upsampling.

4.3 Signal processing attacks

The third and last attack category considered in our experiments was that of signal processing manipulations. A very usual attack is JPEG compression. Figure 20 presents results for quality factor ranging from 10% to 50%. We can notice that the state-of-the-art techniques are superior, with the technique by Tian et al. having the least variation in robustness. However, the performance of our method in all its versions is quite close to the one by Dong et al. especially for high compression ratios (low quality factor values). Of course, for higher quality factor values, the performance of all versions improves since the distortion is smaller. The SIFT-based version of our method is the best, followed by SURF-based, RST-based, and FAST-based.

A more modern compression technique, specifically H.264 intra-frame compression, has also been considered. As we can see in Figure 21, all methods have similar performance which improves with reduced

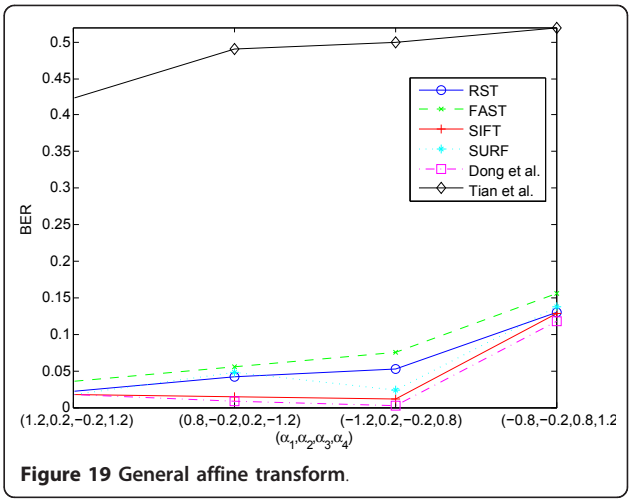


Figure 19 General affine transform.

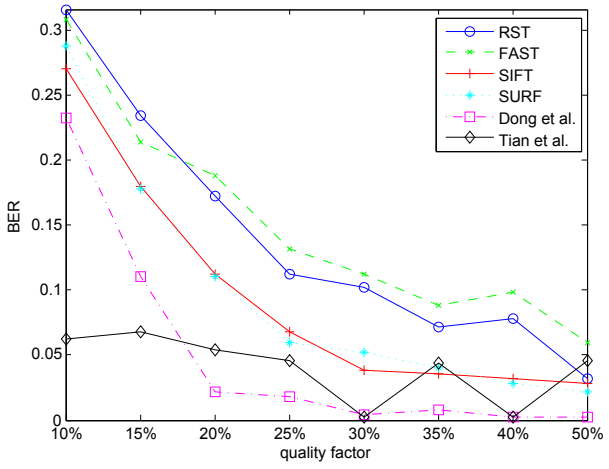


Figure 20 JPEG compression.

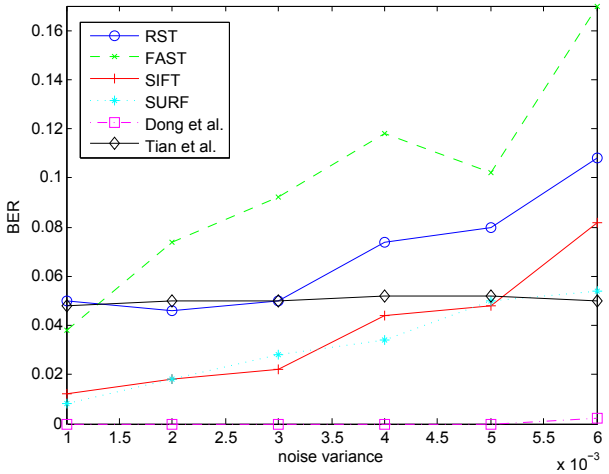


Figure 22 Gaussian noise addition.

quantization parameter value, as expected. The two state-of-the-art techniques perform slightly better, whereas the various versions of our method follow closely, with SIFT-based being the best, followed by SURF-based, RST-based, and FAST-based.

Another common distortion is noise addition. For the purpose of our experiments, we added Gaussian white noise of zero mean value and variance ranging from 0.001 to 0.006 to the watermarked images whose pixel values had previously been scaled to the range [0, 1]. As we can see in Figure 22, our technique is not as robust against Gaussian noise as the technique by Dong et al., but is better than the technique by Tian et al. in its SIFT-based and SURF-based versions. The RST-based version follows and, finally, the FAST-based version exhibits the lowest robustness.

Finally, we perform lowpass filtering using a rotationally symmetric Gaussian filter of size 3 × 3 with standard deviation varying from 0.1 to 0.6. As one can see in Figure 23, the technique by Dong et al. is flawless for all values of standard deviation, followed in order of performance by the SIFT-based and the SURF-based versions of our method, with RST-based and FAST-based following. The technique by Tian et al. is only better than the two latter versions for small values of standard deviation. However, the variation in performance is quite small for all methods.

In summary, the proposed technique, as expected due to its design, is more robust than the state-of-the-art techniques in terms of local geometric distortions. It is also better in terms of shearing attacks and downsampling followed by upsampling. It is only inferior compared to the method by Dong et al., yet with significant performance, under rotation, scaling, general affine transform and signal processing attacks, such as JPEG compression, H.264 intra-frame compression, lowpass

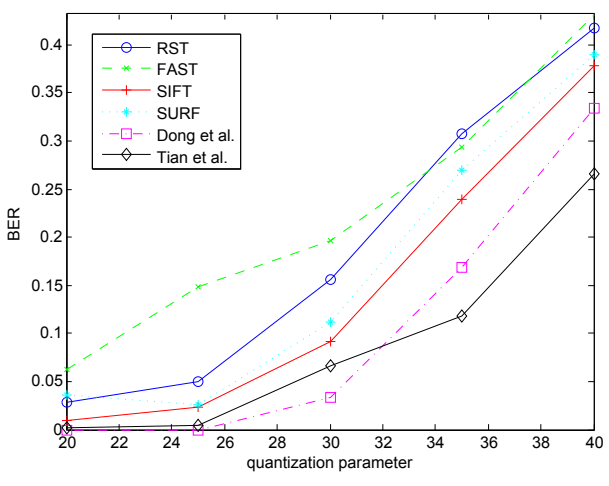


Figure 21 H.264 intra-frame compression.

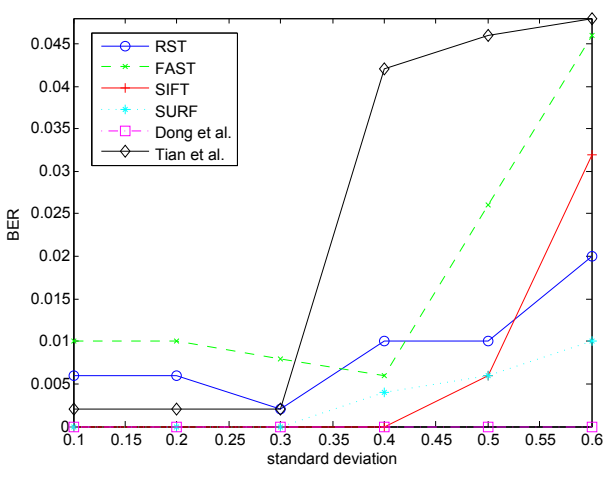


Figure 23 Lowpass filtering.

filtering and noise addition. It is even better, in its SIFT-based and SURF-based versions, than the method by Tian et al. for all these attacks except compression attacks. The most competitive version of our method appears to be the SIFT-based one, followed by the SURF-based, the RST-based, and the FAST-based.

5 Conclusions

In the current article, a new image watermarking technique is proposed, which is robust against the usual local distortion attacks that are not efficiently coped with by the state-of-the-art techniques. According to our technique, a multibit watermark is formed in the DCT domain, inversely transformed and, eventually, geometrically normalized to the spatial domain of the original image. This prevents image interpolation errors in contrast to other techniques in the literature which embed the watermark in a normalized version of the image and afterwards apply inverse normalization. Furthermore, no local search is needed to achieve synchronization during detection. The use of a visibility rule during embedding prevents image deterioration due to overlapping of watermarked areas. Four different feature detection techniques are alternatively used in our study, namely SIFT, SURF, RST, and FAST, in order to produce the regions in which to embed the watermark. Our technique, especially in its SIFT-based version, proves to be more robust against local geometric attacks than certain state-of-the-art techniques and has remarkable performance in terms of global geometric distortions and signal processing attacks.

Acknowledgements

A. Nikolaidis wishes to acknowledge financial support provided by the Research Committee of the Technological Educational Institute of Serres, Greece, under grant SAT/IC/23-3-11-25/1.

Competing interests

The authors declare that they have no competing interests.

Received: 19 October 2011 Accepted: 2 May 2012
Published: 2 May 2012

References

1. JJK O'Ruanaidh, WJ Dowling, FM Boland, Watermarking digital images for copyright protection. IEE Proc Vision Image Signal Process. **143**(4), 250–256 (1996). doi:10.1049/ip-vis:19960711
2. H Berghel, L O'Gorman, Protecting ownership rights through digital watermarking. Computer. **29**(7), 101–103 (1996). doi:10.1109/2.511977
3. IJ Cox, J Kilian, FT Leighton, T Shamoon, Secure spread spectrum watermarking for multimedia. IEEE Trans Image Process. **6**(12), 1673–1687 (1996)
4. W-N Lie, T-L Hsu, G-S Lin, Verification of image content integrity by using dual watermarking on wavelets domain, in *Proc of the IEEE International Conference on Image Processing (ICIP 2003)*, Barcelona, Spain, **3**, 487–490 (2003)
5. D-S Wang, J-P Li, X-Y Wen, Biometric image integrity authentication based on SVD and fragile watermarking, in *Proc of the 2008 Congress on Image and Signal Processing (CISP 2008)*, Sanya, China, **5**, 679–682 (2008)
6. G Depovere, T Kalker, J Haitsma, M Maes, L de Strycker, P Termont, J Vandewege, A Langell, C Alm, P Norman, G O'Reilly, B Howes, H Vaanholt, R Hintzen, P Donnelly, A Hudson, The VIVA project: digital watermarking for broadcast monitoring, in *Proc of the IEEE International Conference on Image Processing (ICIP 1999)*, Kobe, Japan, **2**, 202–205 (1999)
7. L Li, P Daiyuan, L Xiaoju, A Security Video Watermarking Scheme for Broadcast Monitoring, in *Proc, of the 3rd International Workshop on Signal Design and Its Applications in Communications (IWSDA 2007)*, Chengdu, China, **1**, 109–113 (2007)
8. D Kirovski, H Malvar, Y Yacobi, A dual watermark-fingerprint system. IEEE Multimedia. **11**(3), 59–73 (2004). doi:10.1109/MMUL.2004.1
9. Z Shahid, M Chaumont, W Puech, Spread spectrum-based watermarking for Tardos code-based fingerprinting for H.264/AVC video, in *Proc of the IEEE International Conference on Image Processing (ICIP 2010)*, Hong Kong, China, 2105–2108 (2010)
10. I Cox, M Miller, J Bloom, J Fridrich, T Kalker, *Digital Watermarking and Steganography*, 2nd edn. (Morgan Kaufmann, Burlington, MA, 2008)
11. V Solachidis, I Pitas, Circularly symmetric watermark embedding in 2D DFT domain. IEEE Trans Image Process. **10**(11), 1741–1753 (2001). doi:10.1109/83.967401
12. L Verstrepen, T Meesters, T Dams, A Dooms, D Bardyn, Circular Spatial improved watermark embedding using a new Global SIFT synchronization scheme, in *Proc of the 16th International Conference on Digital Signal Processing (DSP 2009)*, Santorini, Greece, **1**, 1–8 (2009)
13. D Zheng, S Wang, J Zhao, RST invariant image watermarking algorithm with mathematical modeling and analysis of the watermarking processes. IEEE Trans Image Process. **18**(5), 1055–1068 (2009)
14. JS Seo, CD Chang, D Yoo, Localized image watermarking based on feature points of scale-space representation. Pattern Recogn. **37**(7), 1365–1375 (2004). doi:10.1016/j.patcog.2003.12.013
15. W Lu, H Lu, F-L Chung, Feature based robust watermarking using image normalization. Comput Electric Eng. **36**(1), 2–18 (2010). doi:10.1016/j.compeleceng.2009.04.002
16. X-Y Wang, Y-P Yang, H-Y Yang, Invariant image watermarking using multi-scale Harris detector and wavelet moments. Comput Electric Eng. **36**(1), 31–44 (2010). doi:10.1016/j.compeleceng.2009.04.005
17. L-D Li, B-L Guo, Localized image watermarking in spatial domain resistant to geometric attacks. AEU - Int J Electron Commun. **63**(2), 123–131 (2009). doi:10.1016/j.aeue.2007.11.007
18. H-Y Lee, H Kim, H-K Lee, Robust image watermarking using local invariant features. Opt Eng. **45**(3), 037002 (2006). doi:10.1117/1.2181887. doi:10.1117/1.2181887
19. P Dong, JG Brankov, NP Galatsanos, Y Yang, F Davoine, Digital Watermarking Robust to Geometric Distortions. IEEE Trans Image Process. **14**(12), 2140–2150 (2005)
20. DG Lowe, Distinctive Image Features from Scale-Invariant Keypoints. Int J Comput Vision. **60**(2), 91–110 (2004)
21. VQ Pham, T Miyaki, T Yamasaki, K Aizawa, Geometrically Invariant Object-Based Watermarking using SIFT Feature, in *Proc of the IEEE International Conference on Image Processing (ICIP 2007)*, San Antonio, Texas, **5**, 473–476 (2007)
22. L Jing, L Gang, Z Jiulong, Robust image watermarking based on SIFT feature and optimal triangulation, in *Proc of the 2009 International Forum on Information Technology and Applications (IFITA 2009)*, Chengdu, China, **3**, 337–340 (2009)
23. J Sun, S Lan, Geometrical attack robust spatial digital watermarking based on improved SIFT, in *Proc of the 2010 International Conference on Innovative Computing and Communication and 2010 Asia-Pacific Conference on Information Technology and Ocean Engineering (CICC-ITOE 2010)*, Macao, Macao, **1**, 98–101 (2010)
24. G Loy, A Zelinsky, Fast radial symmetry for detecting points of interest. IEEE Trans Pattern Anal Mach Intell. **25**(8), 959–973 (2003). doi:10.1109/TPAMI.2003.1217601
25. H Bay, T Tuytelaars, L Van Gool, SURF: Speeded Up Robust Features, in *Proc of the European Conference on Computer Vision (ECCV 2006)*, Graz, Austria, **1**, 404–417 (2006)
26. H Bay, A Ess, T Tuytelaars, L Van Gool, SURF: Speeded Up Robust Features. Comput Vision Image Understand. **110**(3), 346–359 (2008). doi:10.1016/j.cviu.2007.09.014
27. E Rosten, T Drummond, Fusing points and lines for high performance tracking, in *Proc of the 10th IEEE International Conference on Computer Vision (ICCV 2005)*, Beijing, China, **2**, 1508–1511 (2005)

28. E Rosten, T Drummond, Machine learning for high-speed corner detection, in *Proc of the European Conference on Computer Vision (ECCV 2006)*, Graz, Austria, **1**, 430–443 (2006)
29. JF Canny, A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell. **8**(6), 679–698 (1986)
30. AG Bors, I Pitas, Image watermarking using block site selection and DCT domain constraints. Optics Express. **3**(12), 512–522 (1998). doi:10.1364/OE.3.000512
31. S Stankovic, I Orovic, N Zaric, An application of multidimensional time-frequency analysis as a base for the unified watermarking approach. IEEE Trans Image Process. **19**(3), 736–745 (2010)
32. A Nikolaidis, I Pitas, Region-based image watermarking. IEEE Trans Image Process. **10**(11), 1726–1740 (2001). doi:10.1109/83.967400
33. H Tian, Y Zhao, R Ni, J-S Pan, Spread spectrum-based image watermarking resistant to rotation and scaling using radon transform, in *Proc of the Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2010)*, Darmstadt, Germany, **1**, 442–445 (2010)
34. S Pereira, S Voloshynovskiy, M Madueno, S Marchand-Maillet, T Pun, Second generation benchmarking and application oriented evaluation, in *International Workshop on Information Hiding (IHW 2001)*, Pittsburgh, PA, USA, **1**, 340–353 (2001)

doi:10.1186/1687-6180-2012-97
Cite this article as: Nikolaidis: Local distortion resistant image watermarking relying on salient feature extraction. *EURASIP Journal on Advances in Signal Processing* 2012 **2012**:97.