DIGITAL WATERMARKING ROBUST TO GEOMETRIC DISTORTIONS

Ping Dong[1], Jovan G. Brankov[1], Nikolas P. Galatsanos[2,*],
Yongyi Yang[1], and Franck Davoine[3]

[1]Department of Electrical and Computer Engineering
Illinois Institute of Technology
3301 S. Dearborn St., Chicago, IL 60616, USA

[2]Computer Science Department
University of Ioannina
Ioannina, Greece 45110
galatsanos@cs.uoi.gr

[3]Laboratory HEUDIASYC-CNRS
University of Technology of Compiegne
BP 20529—60205 Compiegne, France

ABSTRACT

In this paper we present two watermarking approaches that are robust to geometric distortions. The first approach is based on image normalization, in which both watermark embedding and extraction are carried out with respect to an image normalized to meet predefined moment criteria. We propose a new normalization procedure, which is invariant to affine transform attacks. The resulting watermarking scheme is suitable for public watermarking applications, where the original image is not available for watermark extraction. The second approach is based on a watermark resynchronization scheme aimed to alleviate the effects of random bending attacks. In this scheme, a deformable mesh is used to correct the distortion caused by the attack. The watermark is then extracted from the corrected image. In contrast to the first scheme, the latter is suitable for private watermarking applications, where the original image is needed during watermark detection. In both schemes we employ a direct-sequence code division multiple access (DS-CDMA) approach to embed a multi-bit watermark in the discrete cosine transform (DCT) domain of the image. Numerical experiments demonstrate that the proposed watermarking schemes are robust to a wide range of geometric attacks.

Keywords: Digital watermarking, image normalization, geometric attacks, watermark resynchronization, mesh modeling, code division multiple access watermarking.

[*] Corresponding author

I. INTRODUCTION

With the ever growing expansion of digital multimedia and the Internet the problem of ownership protection of digital information has become increasingly important. Although significant progress has been made in watermarking of digital images, many challenging problems still remain in practical applications. Among these problems is the resilience of watermarking to geometric attacks. Such attacks are easy to implement, but can make many of the existing watermarking algorithms ineffective. Examples of geometric attacks include rotation, scaling, translation, shearing, random bending, or change of aspect ratio (e.g., [1], [2] and [3]). Such attacks are effective in that they can destroy the synchronization in a watermarked bit-steam, which is vital for most of the watermarking techniques. This is problematic especially in applications where multi-bit public watermarking is used, where the original image is not available for watermark extraction.

In the literature several approaches have been proposed to combat geometric attacks. Ruanaidh and Pun [4] proposed a scheme based on the invariant properties of Fourier-Mellin transform (FMT) to deal with such attacks as rotation, scaling and translation (RST). This approach was effective in theory, but difficult to implement. Aimed to alleviate the implementation difficulty of this approach, Lin *et al* [5] proposed to embed the watermark in a one-dimensional signal obtained by projecting the Fourier-Mellin transformed image onto the log-radius axis. This approach was intended to embed only one bit of information, i.e. presence or absence of the watermark.

In [6] Pereira and Pun proposed another approach in which an additional template, known as a "pilot" signal in traditional communication systems, besides the watermark was embedded in the DFT domain of the image. This embedded template was used to estimate the affine geometric attacks in the image. The image was then corrected with the estimated distortion, and the detection of the watermark was performed afterward. A theoretical analysis was provided in [7] on the bit error rate for this pilot-based approach under a number of geometric attacks. This approach requires the detection of both the synchronization pattern and the watermark. A potential problem arises when a common template is used for different watermarked images, making it susceptible to collusion-type detection of the template [8].

In [9] Bas *et al* proposed a watermarking approach that is adaptive to the image content. In this approach salient feature points, extracted from the image, were used to define a number of triangular regions. A one-bit watermark was then embedded inside each triangle using an additive spread spectrum scheme. This approach requires the robust detection of the salient points in the image in order to retrieve the watermark.

In [11] a watermarking scheme was proposed using moment based image normalization, a well-known technique in computer vision and pattern recognition applications [10]. In this approach both watermark embedding and extraction were performed using a normalized image having a standard size and orientation. Thus, it is suitable for public watermarking where the original image is not available. The approach in [11] was used to embed a one-bit watermark.

In this paper, we propose two watermarking approaches to alleviate the problem of geometric distortions. The first is a multi-bit public watermarking scheme based on image normalization, aimed to be robust to general affine geometric attacks. Our scheme is different from the one in [11] in that: 1) we address more general affine distortions, where shearing in the x and y directions are allowed rather than simple scaling and rotation attacks; 2) we propose a multi-bit watermarking system based on direct-sequence code division multiple access (DS-CDMA).

The second watermarking approach is based on a watermark resynchronization scheme, aimed to be robust to random geometric distortions and to be used in the context of private watermarking where the original image is known. This scheme uses a deformable mesh model for correcting the distortion so that resynchronization is achieved. We present and compare two variations of this scheme, which were first reported in our previous work in [13] and [22], respectively.

The rest of this paper is organized as follows. In Section II we present the public watermarking scheme based on image normalization. In Section III we describe the private watermarking scheme based on deformable mesh modeling. In Section IV we present numerical experiments to demonstrate the effectiveness of the proposed algorithms. Finally, we give our conclusions in Section V.

II. WATERMARKING BASED ON IMAGE NORMALIZATION

The key idea of this watermarking scheme is to use a normalized image for both watermark embedding and detection. The normalized image is obtained from a geometric transformation procedure that is invariant to any affine distortions of the image. This will ensure the integrity of the watermark in the normalized image even when the image undergoes affine geometric attacks. This watermarking scheme is illustrated in Figure 1 using a functional diagram. It is noted that the cover image is not needed for the watermark extraction. Thus, this scheme is desirable for public watermarking applications.

Below we describe the components that define this scheme in details. We begin with some background on image moments and geometric affine transforms, which are the necessary tools for image normalization.

A. Image Moments and Affine Transforms

Let $f(x,y)$ denote a digital image of size $M \times N$. Its geometric *moments* m_{pq} and *central moments* μ_{pq}, $p,q = 0,1,2,\cdots$, are respectively defined as

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

and

$$\mu_{pq} = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} (x-\overline{x})^p (y-\overline{y})^q f(x,y), \qquad (2)$$

where

$$\overline{x} = \frac{m_{10}}{m_{00}}, \overline{y} = \frac{m_{01}}{m_{00}}. \qquad (3)$$

An image $g(x,y)$ is said to be an *affine transform* of $f(x,y)$ if there is a matrix $\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ and vector $\mathbf{d} = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$ such that $g(x,y) = f(x_a, y_a)$, where

$$\begin{pmatrix} x_a \\ y_a \end{pmatrix} = \mathbf{A} \cdot \begin{pmatrix} x \\ y \end{pmatrix} - \mathbf{d} . \tag{4}$$

It is readily seen that rotation, scaling, and translation (RST) are all special cases of affine transforms. Other examples of affine transforms include: i) *shearing* in the x direction, which corresponds to $\mathbf{A} = \begin{pmatrix} 1 & \beta \\ 0 & 1 \end{pmatrix} \triangleq \mathbf{A}_x$ in (4); ii) shearing in the y direction, which corresponds to $\mathbf{A} = \begin{pmatrix} 1 & 0 \\ \gamma & 1 \end{pmatrix} \triangleq \mathbf{A}_y$; and iii) *scaling* in both x and y directions, which corresponds to $\mathbf{A} = \begin{pmatrix} \alpha & 0 \\ 0 & \delta \end{pmatrix} \triangleq \mathbf{A}_s$. Moreover, it is straightforward to show that any affine transform $\mathbf{A}$ can be decomposed as a composition of the aforementioned three transforms, e.g., $\mathbf{A} = \mathbf{A}_s \cdot \mathbf{A}_y \cdot \mathbf{A}_x$, provided that $a_{11} \neq 0$ and $\det(\mathbf{A}) \neq 0$.

In addition, one can derive the following results (the derivation is omitted for brevity):

Lemma 1. If $g(x, y)$ is an affine transformed image of $f(x, y)$ obtained with affine matrix $\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ and $\mathbf{d} = \mathbf{0}$, then the following identities hold:

$$m'_{p,q} = \sum_{i=0}^{p} \sum_{j=0}^{q} \binom{p}{i} \binom{q}{j} a_{11}^{i} \cdot a_{12}^{p-i} \cdot a_{21}^{j} \cdot a_{22}^{q-j} \cdot m_{i+j,p+q-i-j} , \tag{5}$$

$$\mu'_{p,q} = \sum_{i=0}^{p} \sum_{j=0}^{q} \binom{p}{i} \binom{q}{j} a_{11}^{i} \cdot a_{12}^{p-i} \cdot a_{21}^{j} \cdot a_{22}^{q-j} \cdot \mu_{i+j,p+q-i-j} , \tag{6}$$

where m'_{pq} , m'_{pq} are the moments of $g(x, y)$, and m_{pq} , m_{pq} are the moments of $f(x, y)$.

B. Image Normalization

In this section, we describe a normalization procedure that achieves invariance under affine geometric distortions. The general concept of image normalization using moments is well-known in pattern recognition problems (e.g., see [15], [16] and [17], where the idea is to extract image features that are invariant to affine transforms). In this application we apply a normalization procedure to the image so that it meets a set of predefined moment criteria.

The normalization procedure consists of the following steps: for a given image $f(x, y)$,

1. Center the image $f(x,y)$; this is achieved by setting in (4) the matrix $\mathbf{A} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and the vector $\mathbf{d} = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$ with

   $$d_1 = \frac{m_{10}}{m_{00}}, d_2 = \frac{m_{01}}{m_{00}}, \tag{7}$$

   where m_{10}, m_{01} and m_{00} are the moments of $f(x,y)$ as defined in (1). This step is aimed to achieve translation invariance. Let $f_1(x,y)$ denote the resulting centered image.

2. Apply a shearing transform to $f_1(x,y)$ in the x direction with matrix $\mathbf{A}_x = \begin{pmatrix} 1 & \beta \\ 0 & 1 \end{pmatrix}$ so that the resulting image, denoted by $f_2(x,y) \triangleq \mathbf{A}_x[f_1(x,y)]$, achieves $\mu_{30}^{(2)} = 0$, where the superscript is used to denote $f_2(x,y)$.

3. Apply a shearing transform to $f_2(x,y)$ in the y direction with matrix $\mathbf{A}_y = \begin{pmatrix} 1 & 0 \\ \gamma & 1 \end{pmatrix}$ so that the resulting image, denoted by $f_3(x,y) \triangleq \mathbf{A}_y[f_2(x,y)]$, achieves $\mu_{11}^{(3)} = 0$.

4. Scale $f_3(x,y)$ in both x and y directions with $\mathbf{A}_s = \begin{pmatrix} \alpha & 0 \\ 0 & \delta \end{pmatrix}$ so that the resulting image, denoted by $f_4(x,y) \triangleq \mathbf{A}_s[f_3(x,y)]$, achieves: 1) a prescribed standard size, and 2) $\mu_{50}^{(4)} > 0$ and $\mu_{05}^{(4)} > 0$.

The final image $f_4(x,y)$ is the normalized image, based on which subsequent watermark embedding or extraction is performed. Intuitively, the above normalization procedure can also be explained as follows: the discussion following equation (4) points to the fact a that a general affine transformation attack can be decomposed as a composition of translation, shearing in both x and y directions, and scaling in both x and y directions. The four steps in the normalization procedure are designed to eliminate each of these distortion components. More specifically step 1 eliminates the translation of the affine attack by setting the center of the normalized image at the density center of the affine attacked image, step 2 and step 3 eliminate shearing in the x and y directions by forcing $\mu_{30}^{(2)} = 0$ and $\mu_{11}^{(3)} = 0$. Finally, step 4 eliminates scaling distortion by forcing the normalized image to a standard size. It is important to note that each step in the normalization procedure is readily invertible. This will allow us to convert the normalized image back to its original size and orientation once the watermark is inserted.

Of course, we need to determine in the above procedure the parameters associated with the transforms $\mathbf{A}_x, \mathbf{A}_y$, and $\mathbf{A}_s$. We will address this issue in the next subsection. In the following theorem we present the invariant property of the normalized image $f_4(x, y)$ to affine transforms.

Theorem 1 . *An image $f(x, y)$ and its affine transforms have the same normalized image.*

The proof of this result is deferred to the Appendix.

To demonstrate this normalization procedure, we show in Figure 2(a) an original image "Lena"; in (b) we show this image after an affine distortion; both of these images yield the same image, shown in Figure 2(c), when the normalization procedure is applied.

C. Determination of the Transform Parameters

In this section we show how to determine the parameters associated with the transforms $\mathbf{A}_x, \mathbf{A}_s$, and $\mathbf{A}_y$ so that they achieve their respective normalization goals.

1. Sheering matrix $\mathbf{A}_x = \begin{pmatrix} 1 & \beta \\ 0 & 1 \end{pmatrix}$.

From identity (6), we have

$$\mu_{30}^{(2)} = \mu_{30}^{(1)} + 3\beta\mu_{21}^{(1)} + 3\beta^2\mu_{12}^{(1)} + \beta^3\mu_{03}^{(1)}, \tag{8}$$

where $\mu_{pq}^{(1)}$ are the central moments of $f_1(x, y)$.

Setting $\mu_{30}^{(2)} = 0$, we obtain

$$\mu_{30}^{(1)} + 3\beta\mu_{21}^{(1)} + 3\beta^2\mu_{12}^{(1)} + \beta^3\mu_{03}^{(1)} = 0. \tag{9}$$

The parameter β is then solved from (9) .

Note that equation (9) can have up to three roots in the case that $\mu_{03}^{(1)} \neq 0$ (which is generally true for most of the nature images). In particular, we may have the following two scenarios: 1) one of the three roots is real and the other two are complex; and 2) all three roots are real. In the former case, we simply set β to be the real root; in the latter case, we pick β to be the median of the three real roots. As demonstrated in the Appendix, such a choice of β is to ensure the uniqueness of the resulting normalized image.

Of course, under some very unusual conditions the number of roots of (9) may vary. For example, when all the moments involved in (9) are zero, it will have infinite number of solutions. This can happen when the image is rotationally symmetric, such as a disk or a ring. We refer to [16] and [17] for more details on general normalization procedures.

2. Sheering matrix $\mathbf{A}_y = \begin{pmatrix} 1 & 0 \\ \gamma & 1 \end{pmatrix}$.

From identity (6), we have

$$\mu_{11}^{(3)} = \gamma \mu_{20}^{(2)} + \mu_{11}^{(2)} . \tag{10}$$

Setting $\mu_{11}^{(3)} = 0$, we obtain

$$\gamma = -\frac{\mu_{11}^{(2)}}{\mu_{20}^{(2)}} . \tag{11}$$

Thus, the parameter γ has a unique solution.

3. Scaling matrix $\mathbf{A}_s = \begin{pmatrix} \alpha & 0 \\ 0 & \delta \end{pmatrix}$.

The magnitudes of scaling parameters α and δ are determined by resizing the image $f_3(x, y)$ to a prescribed standard size in both horizontal and vertical directions. Their signs are determined so that both $\mu_{50}^{(4)}$ and $\mu_{05}^{(4)}$ are positive (which can be changed by flipping either horizontally or vertically).

D. *Effect of the Watermark*

It is noted that for watermark embedding, the normalization is applied with respect to the original image, while for watermark extraction it is applied with respect to the watermarked image. It is thus important to design the watermark signal so that it has minimal effect on the normalized image.

Let $w(x,y)$ denote the watermark signal added to the original image $f(x,y)$. Let $m_{pq}^{(w)}$ denote the moments of $w(x,y)$. Then from (7) one can see that it is desirable to have $m_{00}^{(w)} = m_{10}^{(w)} = m_{01}^{(w)} = 0$, so that $w(x,y)$ has no impact on the centering step of the normalization procedure.

In addition, from Equations (8)-(11) it is desirable to have $m_{pq}^{(w)} = 0$ for $p+q=2$ and 3, so that the watermark does not affect the rest of the normalization transforms. It is assume here that $w(x,y)$ and $f(x,y)$ are statistically independent, so their 2nd and 3rd order central moments are additive.

As will be discussed later, the watermark $w(x,y)$ is a CDMA signal generated from a zero-mean Gaussian or uniform source that is added to the mid-frequency DCT coefficients of the image. As will be seen from our numerical examples, such a watermark nearly satisfies all the desirable properties described above, and will have little impact on the normalized image.

E. *Alternative Normalization Procedures*

The normalization procedure described above consists of a sequence of elementary affine transforms (i.e., shearing and scaling operations). We point out that other transform procedures can also be constructed in a similar fashion to achieve affine-transform invariance in a normalized image. For example, one such procedure is the following

$$\mathbf{A} = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} \alpha & 0 \\ 0 & \delta \end{pmatrix} \begin{pmatrix} 1 & \beta \\ 0 & 1 \end{pmatrix}, \tag{12}$$

which consists of 1) shearing in x-direction, 2) scaling in x and y directions, and 3) rotation by angle ϕ. The parameters in the procedure described in (12) can then be determined by enforcing a set of predefined moments for each step. Interested reader can refer to [15] for details

F. Watermarking Algorithm

The image normalization procedure described above yields a normalized image that is invariant to any affine geometric transforms. It is on this normalized image that we perform watermark embedding and detection. In this paper, we chose to use the spread spectrum based DS-CDMA watermarking scheme [19], which is well-known for its robustness to common signal processing attacks, though other watermarking schemes can be used as well.

(1) Watermark Embedding

The watermark embedding procedure is demonstrated in Figure 3 and summarized as follows: To embed a watermark into an image,

1. Apply the normalization procedure to obtain the normalized image.
2. Create a 2D watermark with the same size as the normalized image. This is accomplished by the following steps: a) Generate M 1-D binary pseudo-random sequences $\mathbf{p}_i$, $i = 1,...,M$, as signature patterns using the private key as seed, where M is the number of bits in the watermark message. Each of these sequences has zero mean and takes values from a binary alphabet {-1,1}; b) Create a 1-D DS-CDMA watermark signature $\mathbf{W}_1$ by modulating the watermark message with the patterns generated in a), i.e. $\mathbf{W}_1 = \sum_{i=1}^{M}(2m_i - 1)\mathbf{p}_i$, where m_i is the ith bit (i.e., 0 or 1) in the watermark message; c) Convert the 1-D signature $\mathbf{W}_1$ into a 2-D signature $\mathbf{W}_2$ in a zigzag scan order; d) Apply the inverse discrete cosine transform (IDCT) to the 2-D signature $\mathbf{W}_2$ to produce w_1.
3. Create a mask image, which is a binary image of the same size as the normalized image. This image has 1's within the support of the normalized image and 0's elsewhere.
4. Generate the watermark signature w from w_1 using the mask image by masking off the boundary area. Signature w is the actual final watermark signature.
5. Apply the inverse of the normalization procedure in Step 1 to the watermark signature w, so that it has the same size as the cover image.

6. The final watermark signature is embedded into the original image additively with desired watermarking strength. This produces the watermarked image.

The whole procedure is equivalent to embedding the watermark signature *w* into the DCT domain of the normalized image. A note is that in this procedure we choose to transform the watermark signature to fit the cover image instead of embedding the watermark into the normalized image. This has the advantage that it avoids any distortion which might otherwise have incurred to the cover image. Another remark is that the masking step (i.e., discarding the part of the watermark signature outside the support of the normalized image) is for the ease of implementation. It will not weaken the correlation property of the watermark signature, because the normalized image is simply zero outside its support.

(2) Watermark Extraction

The following steps are taken to decode the embedded watermark in an image,

1. Apply the normalization procedure to obtain the normalized image.
2. Decode the watermark message in the normalized image. This is accomplished in the following steps: a) Regenerate the watermark patterns $\mathbf{p}_i,\ i=1,\ldots,M$, using the same key and following the same procedure as in step 2 of watermark embedding; b) Apply DCT to the normalized image from Step 1; c) Convert the DCT coefficients where the watermark signature is embedded into a 1-D vector, denoted as $\mathbf{c}_w$, through inverse zigzag scan; d) Decode the watermark message bit-by-bit using a correlation detector. That is, the *i*th bit of the watermark message is decoded as

$$\hat{m}_i = \begin{cases} 1, & \mathrm{corr}(\mathbf{c}_\mathrm{w},\mathbf{p}_\mathrm{i}) > 0 \\ 0, & \text{otherwise,} \end{cases} \tag{13}$$

where $\mathrm{corr}(\mathbf{c}_\mathrm{w},\mathbf{p}_\mathrm{i})$ is the correlation of the two vectors.

III. WATERMARK RESYNCHRONIZATION THROUGH DEFORMABLE MESH MODELING

In practice it may well happen that a watermarked image undergoes a geometric attack that cannot be simply described by RST or more general affine transforms. In such a case it is no longer feasible, if not impossible, to describe the actual image distortion by a global geometric transformation model. Such geometric attacks may cause hardly noticeable perceptual distortion, but can make many existing watermarking algorithms vulnerable.

As an example, in Figure 4(a) we show the Lena image embedded with a watermark; in Figure 4(b) we show this image after attack with StirMark [12]. In Figure 4(c) we show the difference between the two images. In Figure 4(d) we show the effect of this same distortion on a rectangular grid corresponding to the image (dashed—before distortion; solid—after distortion). Indeed, the distortion in the image is barely visible, though the actual geometric distortion is rather severe. The actual attack in this case follows the pattern of an elastic sheet, which is deformed by forces of random magnitude and directions at different locations. Such distortions can easily destroy the synchronization (registration) between the watermark in the attacked image and the detector.

A. Watermarking Scheme based on Mesh Modeling

In this section, we propose to use a deformable mesh model to describe the complex geometric distortion in a watermarked image. The deformable mesh serves as a resynchronization tool between a distorted image and its original image for watermark detection. A functional block diagram of a watermarking system based on such a deformable mesh model is shown in Figure 5. Unlike the scheme in Section II, this watermarking scheme requires the knowledge of the original image. Thus, it is suitable for private watermarking applications.

B. Distortion Correction with a Mesh Model

The concept of mesh modeling is rooted in the field of finite element methods. In a mesh model, the domain of an image is divided into a collection of non-overlapping polygonal patches, called *mesh elements*. In a deformable mesh, the mesh elements are allowed to deform between two image frames (e.g., one before distortion, and the other after distortion). The deformation of a mesh element is through the displacement of its vertices (called *mesh nodes*).

Mesh modeling has recently found many important applications in image processing, including image compression, motion tracking and compensation, image processing through geometric manipulation, and medical image analysis, see for example [14],[23].

(1) Mesh Model of the Image Distortion Field

In the following we assume that we have a pair of images: one is the original image denoted by $f(x,y)$, and the other is $f(x,y)$ underwent a geometric distortion, denoted by $f^{(d)}(x,y)$. We want to characterize the point-wise relative displacement between the two images.

Let vector $\mathbf{d}(\mathbf{p})$ denote the relative displacement of a point $\mathbf{p} \triangleq (x,y)$ in the original image. With a mesh model, we first partition the image domain D is into M non-overlapping mesh elements, denoted by D_m, with $m = 1,2,\ldots,M$. Over each element D_m, we model the displacement $\mathbf{d}(\mathbf{p})$ as:

$$\hat{\mathbf{d}}(\mathbf{p}) = \sum_{n=1}^{N} \varphi_n(\mathbf{p})\mathbf{d}_n \tag{14}$$

where $\mathbf{d}_n$ is the displacement vector at the *n*th element node, and $\varphi_n(\mathbf{p})$ is the interpolation basis function associated with node *n*, and N is the total number of mesh nodes.

In practice, polygonal elements (such as triangles or quadrangles) are usually used in mesh models because of the geometric simplicity and ease of manipulation of these shapes. In this paper triangular mesh elements with liner interpolation basis functions are used in (14).

(2) Determination of the Mesh Deformation

The nodal vectors $\mathbf{d}_n$ in the mesh model in (14) are unknown, and have to be determined from the image data. The basic idea is to displace the mesh nodes so that the two images achieve the best match in terms of their intensity on an element-by-element basis. As a matching criterion the following objective function is used:

$$J = \frac{1}{2}\sum_{m=1}^{M}\left[\int_{D_m}\left(f^{(d)}\left(\mathbf{p}+\hat{\mathbf{d}}(\mathbf{p})\right)-f(\mathbf{p})\right)^2 d\mathbf{p}\right]+\rho E_d, \tag{15}$$

where the first term is the matching error accumulated over all M mesh elements between the two images, the same as the one proposed by Wang and Lee [14]. The second term E_d is used to prevent the mesh from being overly deformed. In this paper we consider two forms of definition for E_d : one is defined on mesh regularity as in [14], which is defined as

$$E_d = \frac{1}{2}\sum_{n=1}^{N}\left\|\mathbf{t}_n\right\|^2, \tag{16}$$

where $\mathbf{t}_n = \sum_{l \in T_n}\left(\mathbf{p}_n - \mathbf{p}_l\right)$, and T_n is the set of all the neighboring nodes of node $\mathbf{p}_n$; the other is defined on deformation regularity, which is defined as

$$E_d = \frac{1}{2}\sum_{n=1}^{N}\left\|\mathbf{d}_n - \overline{\mathbf{d}}_n\right\|^2, \tag{17}$$

where N the total number of mesh nodes in the image, and $\overline{\mathbf{d}}_n$ is the average of the displacement vectors of all the neighboring mesh nodes connected to node n. This term is used to enforce the local smoothness in the distortion field. In what follows we will refer to these two different forms of E_d as variation I and II, respectively.

In (15) ρ is a regularization parameter used to control the trade-off between matching accuracy and deformation regularity. The nodal vectors $\mathbf{d}_n$ are determined by numerical minimization of the objective function in (15). In our experiments, a gradient descent algorithm with a line search was used [18].

Once the nodal vectors $\mathbf{d}_n$ are found, the distortion can be computed for each point in the image according to the deformation model in (14). The distorted image can then be corrected as:

$$\hat{f}\left(\mathbf{p}\right) = f^{(d)}\left(\mathbf{p} + \mathbf{d}\left(\mathbf{p}\right)\right). \tag{18}$$

Afterward watermark detection is performed with respect to this corrected image.

As an example, we show in Figure 4(e) the corrected image from the distorted image in Figure 4(b) using the procedure described above. As in Figure 4(c), the difference between this correct image and the pre-distortion image in Figure 4(a) is shown in Figure 4(f). One can see that the geometric distortion has been corrected effectively in Figure 4(e). The regular mesh structure shown in Figure 4(d) was used, in which mesh nodes were placed regularly every 64 pixels along both dimensions. In addition, the distorted image was extended at the boundaries using the mean image value to avoid the boundary effect during the gradient search step.

IV. EXPERIMENTAL RESULTS

A. Image Normalization Based Watermarking

We present two separate experiments to demonstrate the performance of the proposed watermarking scheme: one on multi-bit watermarking, and the other on 1-bit watermarking. In the first experiment, a 50-bit watermark was embedded into a set of test images (10 of them in total, including “Airplane”, “Boat”, “House”, “Peppers”, “Splash”, “Baboon”, “Couple”, “Lena”, “Elaine” and “Lake”) using the proposed algorithm. The watermarked images were then distorted by a variety of geometric and common signal processing attacks (listed later in detail). The proposed algorithm was applied afterwards to decode the embedded watermark messages in these distorted images. The decoding bit-error rate (BER), defined as the ratio between the number of incorrectly decoded bits and the total number of embedded bits, was then computed and averaged over all the test images.

The second experiment was designed to test the proposed watermarking scheme for detection of the presence or absence of a watermark under geometric attacks: 1) aspect ratio changes of (1.1,1.0), which is test case 3(c) in the distortion list given later; 2) shearing of (5%, 5%), test case 5(f) of the list; 3) general affine transform, test case 6(a) of the list. In this experiment, 20 different watermarks were generated, and embedded into each of the test images separately, resulting in a total of 200 watermarked images; in addition, 20 different white noise patterns were created and added into each of the test images, resulting in a total of 200 invalid watermarked images. These images were then distorted under the 3 geometric attacks. The proposed algorithm was then applied to detect the presence of watermarks in these 600 images.

For comparison, the commercial watermarking software Digimarc ImageBridge was tested using the same images with the same distortions in the second experiment. Intermediate results, such as decoding error rates or test statistic values are not available from this software. However, we can collect overall detection results regarding whether a watermark message as a whole exists or not. A remark here is that Digimarc can still detect the presence of a watermark even when the actual watermark message is no longer decodable. Thus, the detection performance of the two algorithms can be fairly compared, so long as the watermark power is kept at the same level, regardless how many bits are embedded. In our experiment, the watermarking strength was adjusted so that the same signal-to-noise ratio (SNR) was achieved by the two algorithms in the watermarked images for the same test image.

The following is a list of attacks used to distort the images in the experiments (note that not all of them are affine transforms):

1. Line and column removal: (a) (1, 1), (b) (1, 5), (c) (5, 1), (d) (5, 17), and (e) (17, 5), where each pair of numbers indicate the number of columns and rows removed, respectively. The removed columns/rows were equidistant.
2. Scaling by different factors: (a) 0.5, (b) 0.75, (c) 0.9, (d) 1.1, (e) 1.5, and (f) 2.
3. Aspect ratio change: (a) (0.8,1.0), (b) (0.9,1.0), (c) (1.1,1.0), (d) (1.2,1.0), (e) (1.0,0.8), (f) (1.0,0.9), (g) (1.0,1.1), and (h) (1.0,1.2), where each pair of numbers indicate the amount of scaling in the x and y directions, respectively.
4. Rotation with different angles: (a) -15°, (b) -10°, (c) 5°, (d) 25°, (e) 35°, (f) 45°, and (g) 80°.
5. Shearing: (a) (0, 1%), (b) (0, 5%), (c) (1%, 0), (d) (5%, 0), (e) (1%, 1%) and (f) (5%, 5%), where each pair of numbers indicate the amount of shearing in the x and y directions, respectively.
6. General geometric affine transformation with matrix: (a) $\begin{pmatrix} 1.1 & 0.2 \\ -0.1 & 0.9 \end{pmatrix}$, (b) $\begin{pmatrix} 0.9 & -0.2 \\ 0.1 & 1.2 \end{pmatrix}$, and

   (c) $\begin{pmatrix} -1.01 & -0.2 \\ -0.2 & 0.8 \end{pmatrix}$.
7. Horizontal and vertical flipping: (a) horizontal, and (b) vertical.

8. StirMark random bending attack (RBA) [12].
9. Common signal processing attacks: (a) median filtering 2x2, (b) median filtering 3x3, (c) median filtering 4x4, (d) sharpening by kernel $\begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}$, (e) Gaussian filtering by kernel $\frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 1 & 1 \end{pmatrix}$, and (f) frequency mode Laplacian removal (FMLR) attack.
10. JPEG compression with different quality factors: (a) 10, (b) 15, (c) 20, (d) 25, (e) 30, (f) 35, (g) 40, and (h) 50.

The test results from the first experiment are summarized in Tables 1. We see from these results that the proposed algorithm achieves very low decoding BER for all the geometric attacks except StirMark random bending attack (test case 8). It is also robust to filtering attacks (test case 9(b) and 9(c)) except for median filtering.

For the second experiment, the histograms of the values of the test statistic (correlation) used for detection from the 200 watermarked and 200 unwatermarked images are plotted in Figure 6(a), (b) and (c), respectively, for the 3 different geometric attacks detailed above. We notice that the proposed algorithm results in perfect detection for all testing images; the histograms for the watermarked and unwatermarked cases do not overlap. In contrast, Digimarc failed to detect the watermark in all 200 watermarked images and there was no false alarm for the 200 unwatermarked images either after the 3 geometric attacks.

B. Mesh Model Based Watermarking

As test images the Lena and Boat images, respectively, were used. A watermark message with 200 bits was embedded into the mid-frequency DCT coefficients of these images using the CDMA algorithm. The watermarked images were then distorted using the StirMark random bending attack [12]. A series of experiments were performed to test the proposed watermarking system. In all experiments, the original non-watermarked image was used as a reference for the distortion correction; the following different sizes were used for the mesh elements: 32×32 pixels, 64×64 pixels, and 128×128 pixels, respectively. Furthermore, both variations of the penalty term in (16) and (17) were tested; the value of the regularization parameter was chosen empirically for each test.

(1) BER vs. Bending Strength

In this experiment, the watermark strength is fixed at λ=0.5. The test results are shown in Figure 7(a), (b). From these results we can see, for both tested images, as expected that as the bending strength of the attack increases, the BER increase too and that the BER is rather insensitive to the number of mesh elements used. In Figure 7(a), (b) we observe that detection performance is rather robust, for small amounts of bending, to the number of mesh nodes used. Furthermore, detection performance is more sensitive to the number of mesh nodes used for large amounts of bending. However, this is not a very serious concern in most practical applications because large amounts of bending are visible and are not used for attacks.

(2) BER vs. Watermarking Strength.

The bending strength is fixed at 5 in the current experiment, and watermarking strength λ is changed from 0.1 to 1.0. The test results are shown in Figure 8(a), (b). With the proposed correction, zero error decoding can be achieved when the watermarking strength λ is close to 1.0 for both images. From Figure 8 it is appears that detection performance is very sable to the number of mesh nodes used as the power of the watermark changes. These results indicate that the best performance was obtained with mesh elements of 64×64 pixels.

The minimization algorithm requires about 10 seconds per iteration on Pentium 4 at 1.7GHz. This is for image size of 512x512 and regular mesh structure at 64 pixel nodal separation. A typical run takes about 10-20 iterations before useful results are produced.

V. CONCLUSIONS

In the first part of this paper we propose a new public watermarking algorithm, which is robust to general affine geometric transformation attacks. The proposed algorithm achieves its robustness by both embedding and detecting the watermark message in the normalized images. The main result in this part of the paper is Theorem 1 in pp. 7 where we show that the normalized image if properly chosen is invariant to affine transforms. By numerical experiments we demonstrate that the proposed algorithm has very low decoding BER when used with multi-bit watermarks and perfect detection of the presence or absence of the watermark when used with single bit watermarks under various affine attacks. We also compared our algorithm with Digimarc and found it to be superior for affine transform geometric attacks.

We also propose watermark resynchronization scheme based on a mesh model to combat nonlinear geometric attacks. The original image and the potentially attacked watermarked image are used to estimate a mesh model of the unknown geometric distortion. This approach can be only used for private watermarking were the original image is known. Watermark detection is performed using the distorted watermarked images after it has been compensated for the geometric attack. In this paper we tested this approach only against random bending attacks generated by StirMark. Using numerical experiments we demonstrate that the proposed methodology works extremely well. However, the proposed methodology is general can be used for other difficult to correct geometric attacks.

Appendix: Proof of Theorem 1

As pointed out in Section II.A, an affine transform can be decomposed into a composition of the following elementary transforms: 1) translation, 2) shearing in the x direction, 3) shearing in the y direction, and 4) scaling in both x and y directions. Therefore, it is sufficient to demonstrate only that the normalization procedure is invariant to these elementary transforms, i.e., it will yield the same normalized image for a given $f(x,y)$ undergoing each of these elementary transforms.

It is readily seen that the normalization procedure is invariant to the translation transform. This is because any translation in $f(x,y)$ is removed by first centering the image in the normalization procedure.

Next, we demonstrate the normalized image of $f(x,y)$ is invariant for each of the other three elementary transforms. Without loss of generality, we will assume that $f(x,y)$ is already centered. We will use $g(x,y)$ to denote the distorted image from $f(x,y)$ after an affine transform. In addition, we will use $\mu_{pq}^{(a)}$ and μ_{pq} to denote the moments of $g(x,y)$ and $f(x,y)$, respectively.

From the normalization procedure described in Section II.B, the normalized image of $g(x,y)$ can be written as $g(x_n, y_n)$, where

$$\begin{pmatrix} x_n \\ y_n \end{pmatrix} = \mathbf{A}_s \mathbf{A}_y \mathbf{A}_x \begin{pmatrix} x_a \\ y_a \end{pmatrix}. \qquad (.19)$$

The parameter β in the matrix $\mathbf{A}_x$ in (.19) is solved from the normalization condition in (8), i.e.,

$$\mu_{30}^{(a)} + 3\beta\mu_{21}^{(a)} + 3\beta^2\mu_{12}^{(a)} + \beta^3\mu_{03}^{(a)} = 0. \qquad (.20)$$

Also, the parameter γ in the matrix $\mathbf{A}_y$ in (.19) is solved from the normalization condition in (11), i.e., it is determined as

$$\gamma = -\frac{\mu_{11}^{(2)}}{\mu_{20}^{(2)}} = -\frac{\mu_{11}^{(1)} + \beta\mu_{02}^{(1)}}{\mu_{20}^{(1)} + 2\beta\mu_{11}^{(1)} + \beta^2\mu_{02}^{(1)}}. \qquad (.21)$$

1. Shearing in the x direction

In this case, $g(x,y) = f(x_a, y_a)$, where

$$\begin{pmatrix} x_a \\ y_a \end{pmatrix} = \mathbf{A}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & \beta_0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}. \qquad (.22)$$

Based on this relation we can write the moments $\mu_{pq}^{(a)}$ in the normalization condition in (.20) in terms of μ_{pq} using (6), and, after some algebra, we can rewrite (.20) as

$$\mu_{30} + 3(\beta+\beta_0)\mu_{21} + 3(\beta+\beta_0)^2\mu_{12} + (\beta+\beta_0)^3\mu_{03} = 0. \qquad (.23)$$

Let $\beta' \triangleq \beta_1 + \beta_2$. One can see that β' satisfies the equation of the shearing parameter β for normalizing the original image $f(x,y)$. Let $\mathbf{A}'_x$ denote the corresponding shearing transform, that is, $\mathbf{A}'_x = \begin{pmatrix} 1 & \beta' \\ 0 & 1 \end{pmatrix}$.

Observe that

$$\mathbf{A}_x\mathbf{A} = \begin{pmatrix} 1 & \beta \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & \beta_0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & \beta+\beta_0 \\ 0 & 1 \end{pmatrix} = \mathbf{A}'_x . \qquad (.24)$$

Thus, the shearing normalization on $g(x,y)$ using will yield the same image as the shearing normalization transform on $f(x,y)$.

2. Shearing in the y direction

In this case, $g(x,y) = f(x_a, y_a)$, where

$$\begin{pmatrix} x_a \\ y_a \end{pmatrix} = \mathbf{A}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \gamma_0 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}. \qquad (.25)$$

As above, we write the moments $\mu_{pq}^{(a)}$ in (.20) in terms of μ_{pq}, and rewrite the normalization condition as

$$\mu_{30} + 3\left(\frac{\beta}{1+\beta\gamma_0}\right)\mu_{21} + 3\left(\frac{\beta}{1+\beta\gamma_0}\right)^2 \mu_{12} + \left(\frac{\beta}{1+\beta\gamma_0}\right)^3 \mu_{03} = 0 . \qquad (.26)$$

Let $\beta' \triangleq \dfrac{\beta}{1+\beta\gamma_0}$. One can see that β' satisfies the equation of the shearing parameter β for normalizing the original image $f(x,y)$.

Next, we write the moments $\mu_{pq}^{(a)}$ in the normalization condition in (.21) in terms of μ_{pq}, and obtain

$$\gamma = -\frac{\gamma_0(1+\beta\gamma_0)\mu_{20} + (1+2\beta\gamma_0)\mu_{11} + \beta\mu_{02}}{(1+\beta\gamma_0)^2\mu_{20} + 2\beta(1+\beta\gamma_0)\mu_{11} + \beta^2\mu_{02}} . \qquad (.27)$$

Hence,

$$\mathbf{A}_y\mathbf{A}_x\mathbf{A} = \begin{pmatrix} 1 & 0 \\ \gamma & 1 \end{pmatrix}\begin{pmatrix} 1 & \beta \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ \gamma_0 & 1 \end{pmatrix} = \begin{pmatrix} 1+\beta\gamma_0 & 0 \\ \gamma + \gamma_0(1+\beta\gamma) & 1+\beta\gamma \end{pmatrix}. \qquad (.28)$$

Upon some algebraic manipulation, (.28) can be rewritten as

$$\mathbf{A}_y\mathbf{A}_x\mathbf{A} = \begin{pmatrix} 1+\beta\gamma_0 & 0 \\ 0 & \dfrac{1+\beta\gamma_0}{(1+\beta\gamma_0)^2\mu_{20} + 2\beta(1+\beta\gamma_0)\mu_{11} + \beta^2\mu_{02}} \end{pmatrix} \begin{pmatrix} 1 & \beta' \\ -\mu_{11} - \beta'\mu_{02} & \mu_{20} + \beta'\mu_{11} \end{pmatrix}. \qquad (.29)$$

Observe the following: 1) the second matrix term in (.29) corresponds to an affine transform that is independent of the parameter γ_0; and 2) the first matrix term in (.29) corresponding to a scaling transform, which will be later absorbed into the scaling matrix $\mathbf{A}_s$ in (.19) to achieve a standard size. Therefore, the resulting normalized image of $g(x, y)$ is invariant to the affine transformation $\mathbf{A}$ which is parameterized by γ_2.

3. Scaling in both x and y directions

In this case, $g(x, y) = f(x_a, y_a)$, where

$$\begin{pmatrix} x_a \\ y_a \end{pmatrix} = \mathbf{A}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \alpha_0 & 0 \\ 0 & \delta_0 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}. \qquad (.30)$$

Again, we write the moments $\mu_{pq}^{(a)}$ in terms of μ_{pq}, and rewrite the normalization condition (.20) as

$$\mu_{30} + 3\left(\frac{\delta_0}{\alpha_0}\beta\right)\mu_{21} + 3\left(\frac{\delta_0}{\alpha_0}\beta\right)^2\mu_{12} + \left(\frac{\delta_0}{\alpha_0}\beta\right)^3\mu_{03} = 0. \qquad (.31)$$

Let $\beta' \triangleq \dfrac{\delta_0}{\alpha_0}\beta$. One can see that β' satisfies the equation of the shearing parameter β for normalizing the original image $f(x, y)$.

Next, we write the moments $\mu_{pq}^{(a)}$ in the normalization condition (.21) in terms of μ_{pq}, and obtain

$$\gamma = -\frac{\alpha_0\delta_0\mu_{11} + \beta\delta_0^2\mu_{02}}{\alpha_0^2\mu_{20} + 2\beta\alpha_0\delta_0\mu_{11} + \beta^2\delta_0^2\mu_{02}}. \qquad (.32)$$

Hence,

$$\mathbf{A}_y\mathbf{A}_x\mathbf{A} = \begin{pmatrix} 1 & 0 \\ \gamma & 1 \end{pmatrix}\begin{pmatrix} 1 & \beta \\ 0 & 1 \end{pmatrix}\begin{pmatrix} \alpha_0 & 0 \\ 0 & \delta_0 \end{pmatrix} = \begin{pmatrix} \alpha_0 & \beta\delta_0 \\ \alpha_0\gamma & \delta_0(1+\beta\gamma) \end{pmatrix}. \qquad (.33)$$

Upon some algebraic manipulation, (.33) can be rewritten as

$$\mathbf{A}_y\mathbf{A}_x\mathbf{A} = \begin{pmatrix} \alpha & 0 \\ 0 & \dfrac{\alpha_0^2\delta}{\alpha_0^2\mu_{20} + 2\beta\alpha_0\delta_0\mu_{11} + \beta^2\delta_0^2\mu_{02}} \end{pmatrix} \begin{pmatrix} 1 & \beta' \\ -\mu_{11} - \beta'\mu_{02} & \mu_{20} + \beta'\mu_{11} \end{pmatrix}. \qquad (.34)$$

Again, the second matrix term in (.34) corresponds to an affine transform that is independent of the parameters α_0, δ_0; and 2) the first matrix term in (.34) corresponding to a scaling transform. Therefore, the resulting normalized image of $g(x, y)$ is invariant to the affine transformation $\mathbf{A}$ which is parameterized by α_0, δ_0.

4. Uniqueness under a general affine transform

Finally, consider the case that the image $f(x, y)$ undergoing a general affine transformation. We decompose the transform matrix $\mathbf{A}$ as

$$\mathbf{A} = \begin{pmatrix} \alpha_0 & 0 \\ 0 & \delta_0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \gamma_0 & 1 \end{pmatrix} \begin{pmatrix} 1 & \beta_0 \\ 0 & 1 \end{pmatrix}. \qquad (.35)$$

Similar to (.23), (.26) and (.31), we can derive, $\beta' = \dfrac{1}{\dfrac{\alpha_0}{\delta_0\beta} + \gamma_0} + \beta_0$, where β' is a root of the normalization condition (.20) that corresponds to the original image $f(x, y)$, and β is roots corresponds to the affine transformed image. Therefore,

$$\beta = \frac{\dfrac{\alpha_0}{\delta_0}}{\dfrac{1}{\beta' - \beta_0} - \gamma_0}. \qquad (.36)$$

From (.36) we see that β is real if and only if β' is real. Thus, if (.20) has only one real root (or three real roots) for the original image $f(x, y)$, then it also has only one real root (or three real roots) for any of its affine transforms.

Furthermore, β is a monotonic function of β' for $|\beta'| < \left| \dfrac{1}{\gamma_0} + \beta_0 \right|$. In such a case, if β' has three real roots, then its median will correspond to the median of β.

We note that the condition that $|\beta'| < \left|\frac{1}{\gamma_0} + \beta_0\right|$ is not restrictive in practice. For example, for meaningful distortions, we will likely have $|\beta_0| < 0.2$, and $|\gamma_0| < 0.2$ (less than 20% shearing in the x-direction or y-direction). In such a case, $\left|\frac{1}{\gamma_0} + \beta_0\right| > 4.8$. This, of course, leaves enough room for the shearing parameter β'.

References

[1] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Attacks on copyright marking systems," in *Workshop on Information Hiding, Portland, OR*, 15-17 April, 1998.

[2] M. Kutter and F. A. P. Petitcolas, "A fair benchmark for image watermarking systems," *Electronic Imaging' 99, Security and Watermarking of Multimedia Contents*, vol.3657, Sans Jose, CA, Jan. 1999.

[3] I. J. Cox and J. P. M. G. Linnartz, "Public watermarks and resistance to tampering," *IEEE International Conference on Image Processing*, vol.3, 1997.

[4] J. O'Ruanaidh and T. Pun, "Rotation, Scale and translation invariant spread spectrum digital image watermarking," *Signal Processing*, vol. 66, no. 3, pp. 303-317, 1998.

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

[6] S. Pereira and T. Pun, "Robust template matching for affine resistant image watermarks," in *IEEE Trans. on Image Processing*, vol. 9, no. 6, pp. 1123-1129, June 2000.

[7] Manuel Alvarez-Rodríguez and Fernando Pérez-González, "Analysis of pilot-based synchronization algorithms for watermarking of still images". *Signal Processing: Image Communication*, 17(8): 661-633, Sept. 2002.

[8] I. J. Cox, M. L. Miller, J. A. Bloom, *Digital Watermarking*, Morgan Kaufmann publishers, 2001.

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

[10] J. Wood, "Invariant pattern recognition: a review," *Pattern Recognition*, vol. 29, no. 1, pp. 1-17, 1996.

[11] M. Alghoniemy and A. H. Tewfik, "Geometric distortion correction through image normalization," *Multimedia and Expo*, ICME 2000.

[12] F. A. P. Petitcolas. StirMark 3.1 1999, available at site: http://www.cl.cam.ac.uk/fapp2/watermarking/stirmark/

[13] F. Davoine, "Triangular meshes: a solution to resist to geometric distortions based watermark-removal softwares," *European Signal Processing Conference*, Tampere, Finland, 2000.

[14] Y. Wang and O. Lee, "Active mesh--a feature seeking and tracking image sequence representation scheme," *IEEE Trans. Image Proc.*, vol. 3, no.5, pp. 610-624, Sept. 1994.

[15] I. Rothe, H. Susse, and K. Voss, "The method of normalization to determine invariants," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no.4, pp.366-376, Apr. 1996.

[16] D. Shen and H. S. Ip, "Generalized affine invariant image normalization," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp.431-440, May 1997.

[17] D. Shen, H.S. Ip, K. K. T. Cheung, and E. K. Teoh, "Symmetry detection by generalized complex (GC) moments: a close-form solution," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp.466-476, May 1999.

[18] S. Nash and A. Sofer, *Linear and Nonlinear Programming*, McGraw Hill 1996.

[19] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. on Image Processing*, vol. 6, no. 12, pp.1673-1687, 1997.

[20] J. L. Dugelay and F. A. P. Petitolas, "Possible counter-attacks against random geometric distortions," *Proceedings of the SPIE Conference on Security and Watermarking of Multimedia Content II*, vol.3971, pp. 338-345, 2000.

[21] P. Dong and N. P. Galatsanos, "Affine transformation resistant watermarking based on image normalization," *IEEE International Conference on Image Processing*, Rochester, NY, Sept. 2002

[22] P. Dong, J. Brankov, N. P. Galatsanos, and Y. Yang, "Geometric robust watermarking through mesh model based correction," *IEEE International Conference on Image Processing*, Rochester, NY, Sept. 2002

[23] K. Aizawa and T. S. Huang, "Model-based image coding: advanced video coding techniques for very low bit-rate applications," *Proc. of IEEE*, vol. 83, no.2, pp. 259-271, Feb. 1995.

[24] Y. Altunbasak and A. M. Tekalp, "Closed-form connectivity-preserving solutions for motion compensation using 2-d meshes," *IEEE Trans. Image Proc.,* vol. 6, no. 9, pp. 1255-1269, Sept. 1997.

[25] A. Singh, D. Goldgof, and D. Terzopoulos, ed., *Deformable Models in Medical Image Analysis*, IEEE Computer Society Press, 1998.

[26] J. G. Brankov, Y. Yang, and M. N. Wernick, "4D processing of gated SPECT images using deformable mesh modeling," *The Sixth International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, Pacific Grove, California, Oct. 30 - Nov. 2, 2001.

[27] Y. Yang, M. Wernick and J. G. Brankov,, "A fast algorithm for accurate content-adaptive mesh generation," *IEEE Trans. on Image Processing,* vol. 12, no. 8, pp.866-881, Aug. 2003.

[28] C. Podilchuk, W. Zeng, "Image adaptive watermarking using visual models," *IEEE Journal Selected Areas of Communications*, vol.16, no.4, pp. 525-539, May 1998.

[29] *M. Kutter and S. Winkler,* "A vision-based masking model for spread-spectrum image watermarking," *IEEE Trans. Image Proc.*, vol. 11, no.1, pp. 16-25, Jan. 2002.

Table 1:Decoding performance of the proposed algorithm (in BER)

Attacks\Cases	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
1. Removal	0	0.004	0	0.004	0			
2. Scaling	0	0	0	0	0	0.048		
3. Aspect ratio	0	0	0	0	0	0	0	0
4. Rotation	0	0	0	0	0	0	0	
5. Shearing	0	0	0	0	0.002	0		
6. Linear transform.	0	0	0					
7. Flip	0	0						
8. StirMark RBA	0.506							
9. Common signal processing	0.066	0.23	0.232	0.064	0	0.018		
10. JPEG	0.052	0.052	0.004	0.006	0.004	0.004	0	0

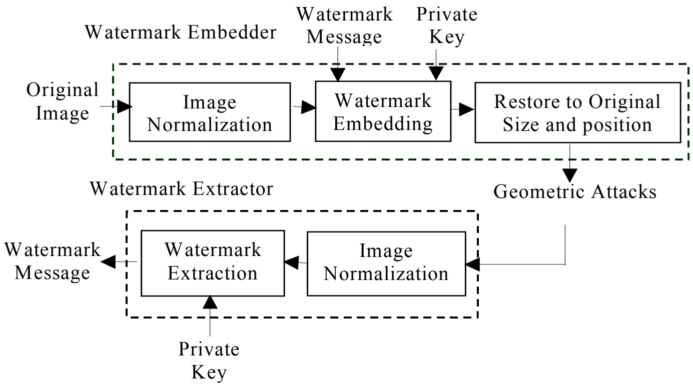


Figure 1. Image normalization based watermarking system.

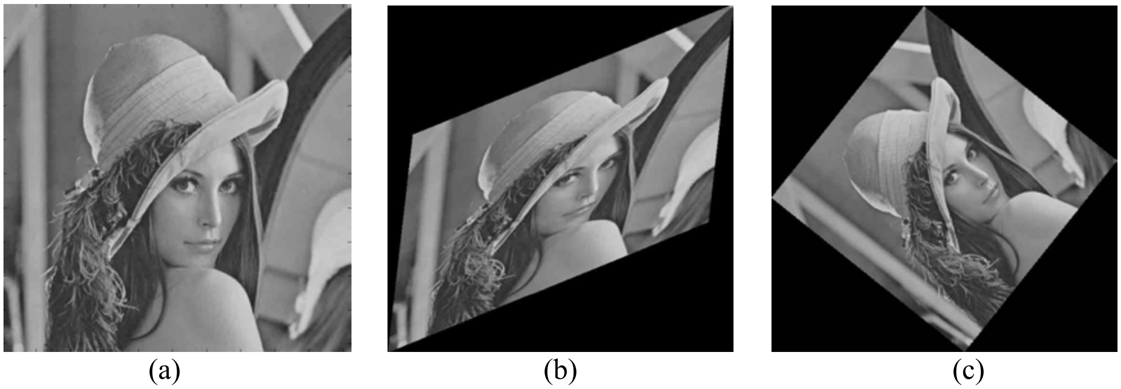

(a) (b) (c)

Figure 2: (a) Original Lena image; (b) Lena image in (a) after distortion; (c) Normalized image from both (a) and (b).

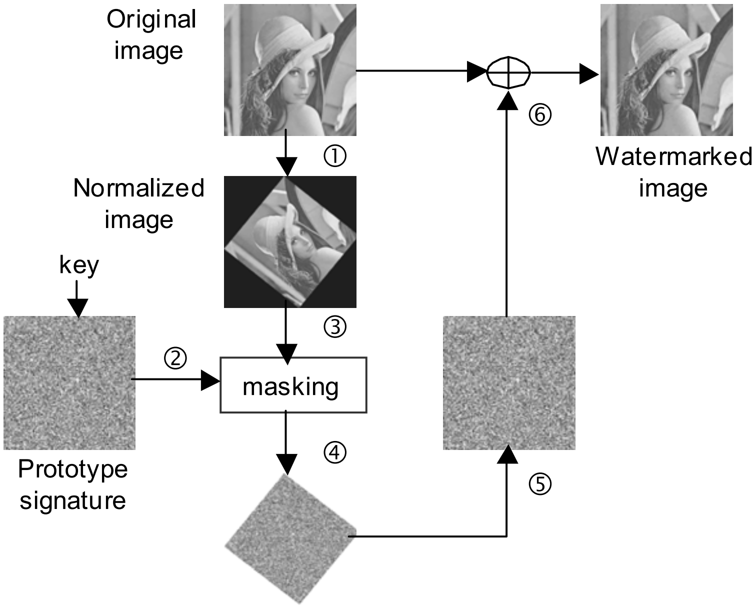
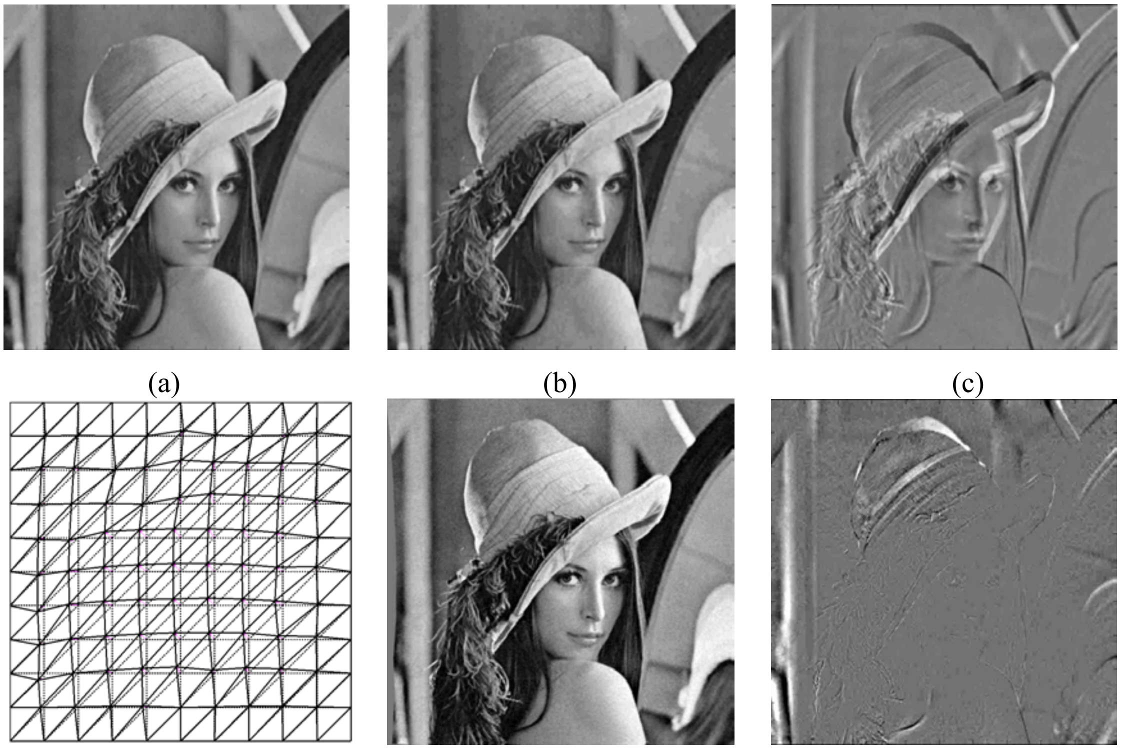


Figure 3: Illustration of watermark embedding process.

(①~⑥ indicates step 1 ~ step 6)



(a) (b) (c)

(d) (e) (f)

.

Figure 4: Images to demonstrate the watermarking process; (a) Watermarked image with PSNR=38.4dB (b) Attacked watermarked image (c) Difference between (a) and (b) (d) Regular mesh and mesh generated from (b) (e) Deformation compensated watermarked image (g) Difference between (a) and (f).

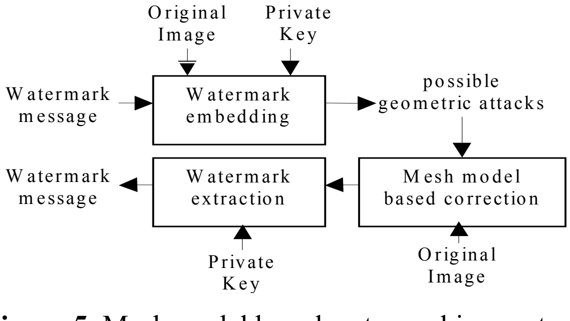
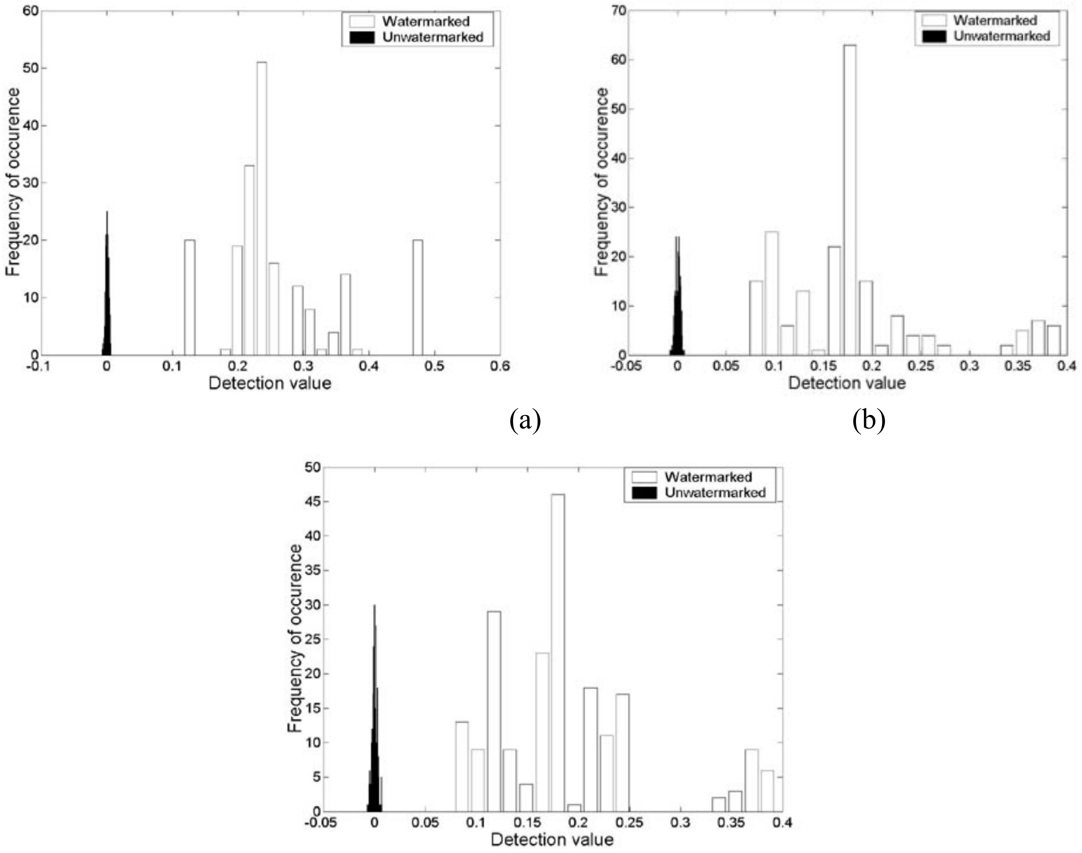


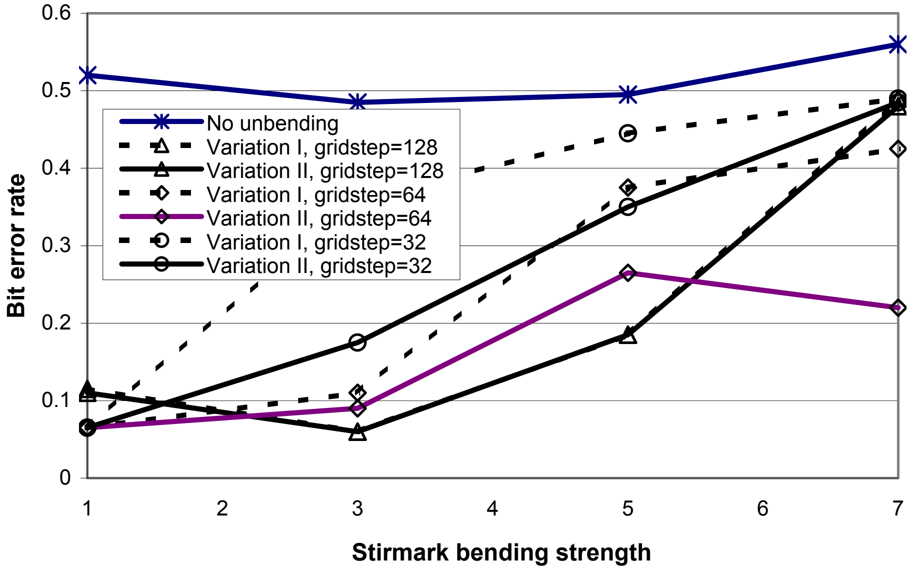
Figure 5. Mesh model based watermarking system



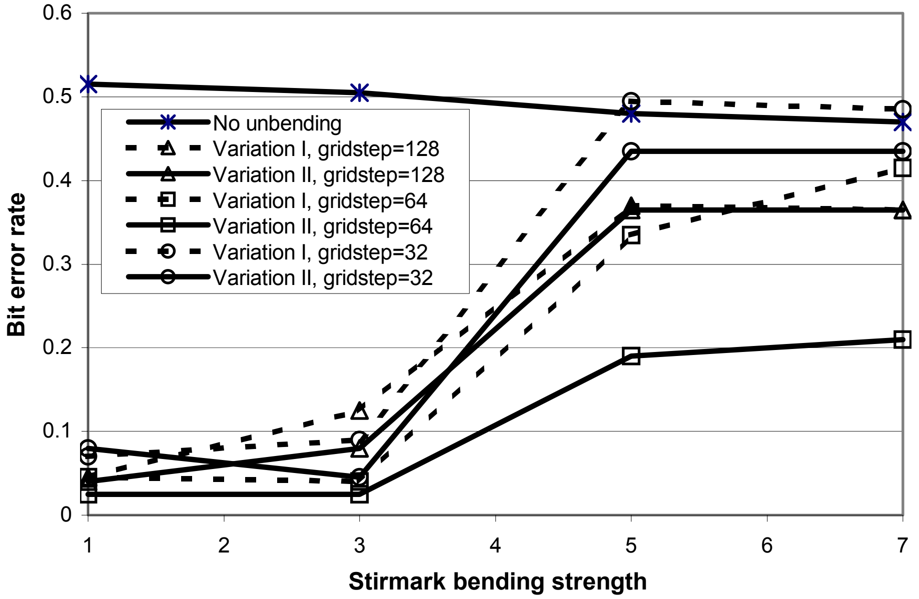
(a) (b)

(c)

Figure 6. Histogram of the values of the test statistic (Normalized cross correlation) used for detection (a) under aspect ratio change (b) under shearing geometric (c) under general affine transformation attacks for 200 watermarked images (left) and 200 unwatermarked images (right).

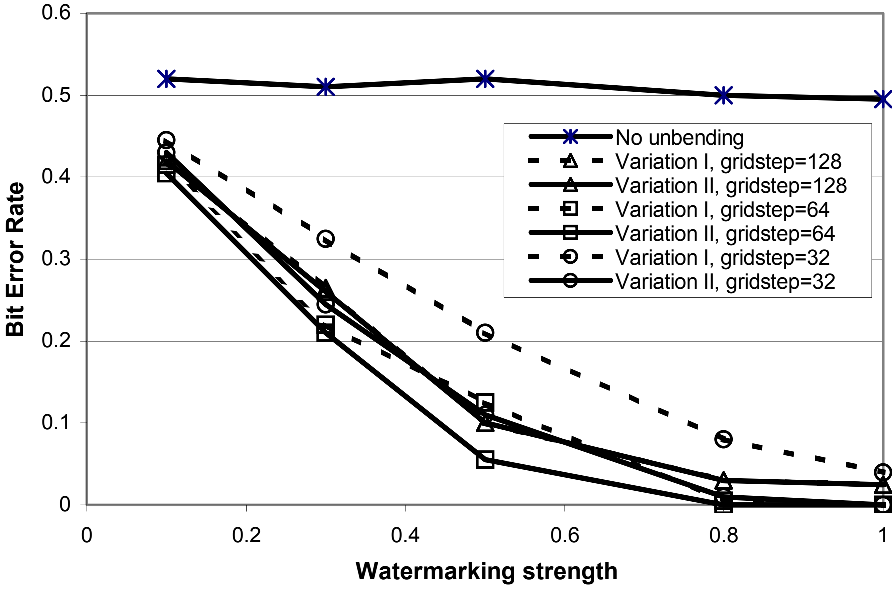


(a)

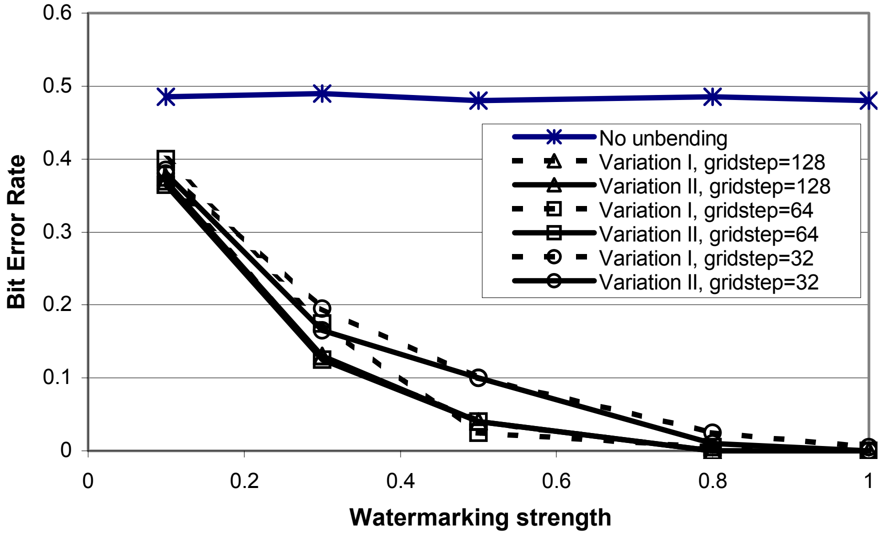


(b)

Figure 7: BER vs. random bending strength (a) Lena (b) Boat.
Note: gridstep of 64 means the mesh nodes are placed 64 pixels apart uniformly.



(a)



(b)

Figure 8: BER vs. watermark strength (a) Lena (b) Boat