Digital Watermarking Robust to Geometric Distortions

Ping Dong, Jovan G. Brankov, *Member, IEEE*, Nikolas P. Galatsanos, *Senior Member, IEEE*, Yongyi Yang, *Senior Member, IEEE*, and Franck Davoine, *Member, IEEE*

***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 a set of 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 necessary for watermark detection. In both schemes, we employ a direct-sequence code division multiple access approach to embed a multibit watermark in the discrete cosine transform domain of the image. Numerical experiments demonstrate that the proposed watermarking schemes are robust to a wide range of geometric attacks.**

***Index Terms*—Code division multiple access (CDMA) watermarking, digital watermarking, geometric attacks, image normalization, mesh modeling, watermark resynchronization.**

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, and change of aspect ratio (e.g., [1]–[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 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 attacks such 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 (1-D) 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 first corrected with the estimated distortion, and the detection of the watermark was performed afterwards. 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 1-bit watermark was then embedded inside each triangle using an additive spread spectrum scheme. This approach requires 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 1-bit watermark.

In this paper, we propose two watermarking approaches to alleviate the problem of geometric distortions. The first is a multibit 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 and 2) we use a multibit watermarking system based on direct-sequence code division multiple access (DS-CDMA).

Manuscript received May 31, 2003; revised October 5, 2004. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Gaurav Sharma.

P. Dong, J. G. Brankov, and Y. Yang are with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL 60616 USA.

N. P. Galatsanos is with the Computer Science Department, University of Ioannina, Ioannina, Greece 45110 (e-mail: galatsanos@cs.uoi.gr).

F. Davoine is with the Laboratory HEUDIASYC-CNRS, Compiegne University of Technology, BP 20529—60205 Compiegne, France.

Digital Object Identifier 10.1109/TIP.2005.857263

The second watermarking approach that we propose 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 to correct 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. A functional diagram of this watermarking scheme is illustrated in Fig. 1. It is noted that the cover image is not needed for the watermark extraction. Thus, this scheme is suitable for public watermarking applications.

We describe the components that define this scheme in detail. 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* $\mu_{pq}, p, q = 0,1,2,\ldots$ are defined, respectively, as

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

and

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

where

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}. \tag{3}$$

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 RST are all special cases of affine transforms. Other examples of affine transforms include: 1) *shearing* in the x direction, which corresponds to $\mathbf{A} = \begin{pmatrix} 1 & \beta \\ 0 & 1 \end{pmatrix} \triangleq \mathbf{A}_x$ in (4); 2) shearing in the y direction, which corresponds to $\mathbf{A} = \begin{pmatrix} 1 & 0 \\ \gamma & 1 \end{pmatrix} \triangleq \mathbf{A}_y$; and 3) *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 $\det(\mathbf{A}) \neq 0$.

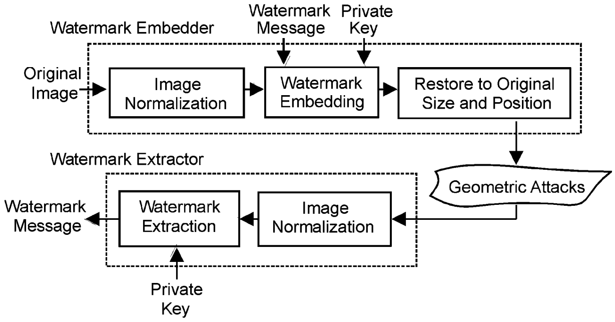


Fig. 1. Image normalization-based watermarking system.

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:

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

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

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]–[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

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

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.

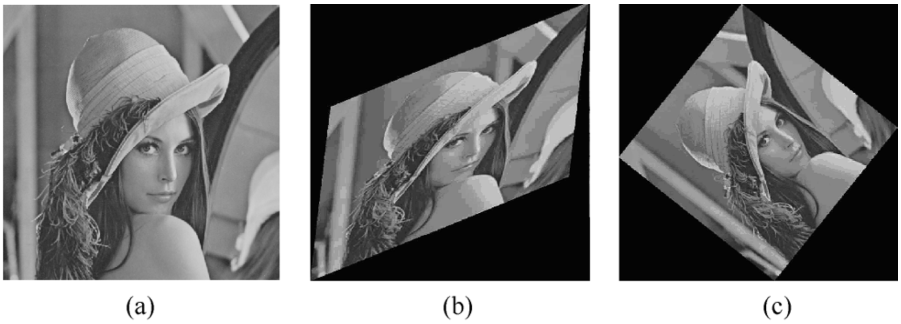


Fig. 2. (a) Original Lena image. (b) Lena image in (a) after distortion. (c) Normalized image from both (a) and (b).

2) Apply a shearing transform to $f_1(x,y)$ in the x direction with matrix $\mathbf{A}_x = (\begin{smallmatrix} 1 & \beta \\ 0 & 1 \end{smallmatrix})$ 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 $\mathbf{A}_y = (\begin{smallmatrix} 1 & 0 \\ \gamma & 1 \end{smallmatrix})$ 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 $\mathbf{A}_s = (\begin{smallmatrix} \alpha & 0 \\ 0 & \delta \end{smallmatrix})$ 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 (4) points to the fact 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, steps 2) and 3) eliminate shearing in the x and y directions, and, 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 Fig. 2(a) an original image "Lena." In Fig. 2(b), we show this image after an affine distortion; both of these images yield the same normalized image, shown in Fig. 2(c), when the above 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 $\mathbf{A}_x = (\begin{smallmatrix} 1 & \beta \\ 0 & 1 \end{smallmatrix})$.

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 found from (9).

Note that (9) can have up to three roots in the case that $\mu_{03}^{(1)} \neq 0$ (which is generally true for most natural 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, this choice of β ensures 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 $\mathbf{A}_y = (\begin{smallmatrix} 1 & 0 \\ \gamma & 1 \end{smallmatrix})$.

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 $\mathbf{A}_s = \left(\begin{smallmatrix}\alpha & 0\\ 0 & \delta\end{smallmatrix}\right)$.

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. Thus, it is 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 $m_{10}^{(w)} = m_{01}^{(w)} = 0$, so that $w(x,y)$ has no impact on the centering step of the normalization procedure.

In addition, from (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 assumed here that $w(x,y)$ and $f(x,y)$ are statistically independent, so their second- and third-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 the 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 readers 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, even though other watermarking schemes can be used as well.

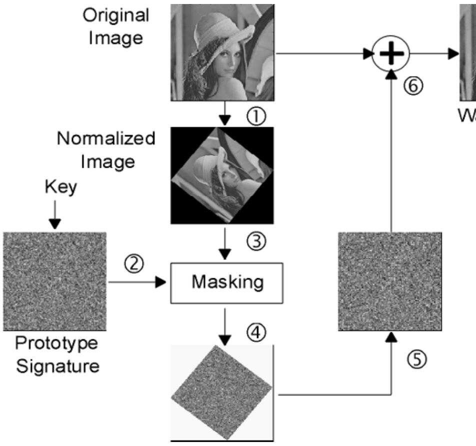


Fig. 3. Illustration of watermark embedding process. [①∼ ⑥indicates step 1) ∼ step 6)].

1) Watermark Embedding: The watermark embedding procedure is demonstrated in Fig. 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 two-dimensional (2-D) 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 $\mathbf{p}_i, i = 1,\ldots,M$, 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 pre-selected zigzag scan (e.g., mid-range DCT coefficients). 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 1s within the support of the normalized image and 0s 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. We note here 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

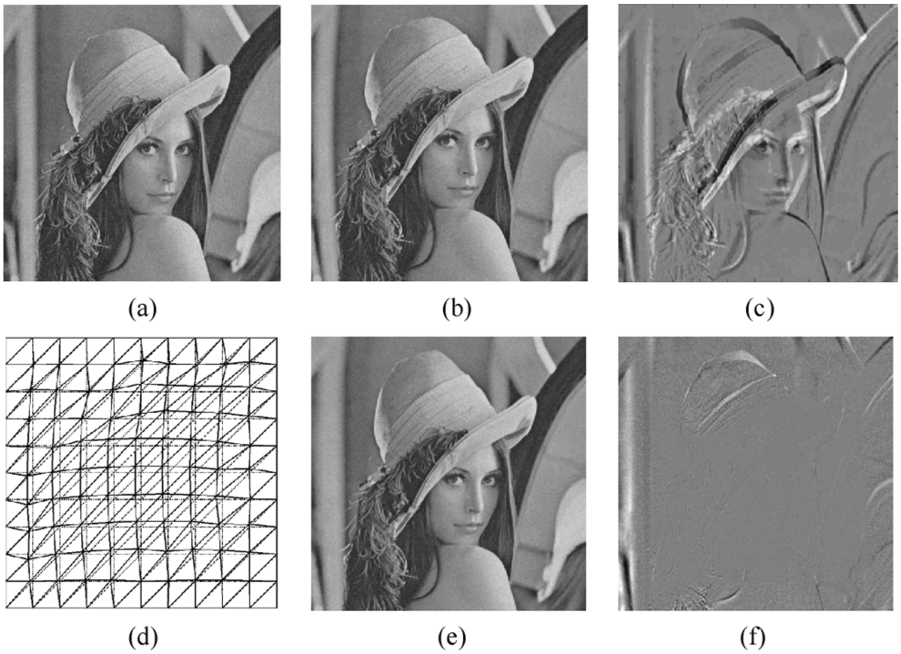


Fig. 4. Images to demonstrate the watermarking process. (a) Watermarked image with $\mathrm{PSNR} = 38.4$ dB. (b) Attacked watermarked image. (c) Difference between (a) and (b). (d) Regular mesh and mesh generated from (b). (e) Deformation compensated watermarked image. (g) Difference between (a) and (e).

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

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

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

III. Watermark Resynchronization Through Deformable Mesh Modeling

In practice, it may very 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 have catastrophic effects to many existing watermarking algorithms.

As an example, in Fig. 4(a), we show the Lena image embedded with a watermark; in Fig. 4(b), we show this image after attack with StirMark [12]. In Fig. 4(c), we show the difference between the two images. In Fig. 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 that at 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 Fig. 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 nonoverlapping 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

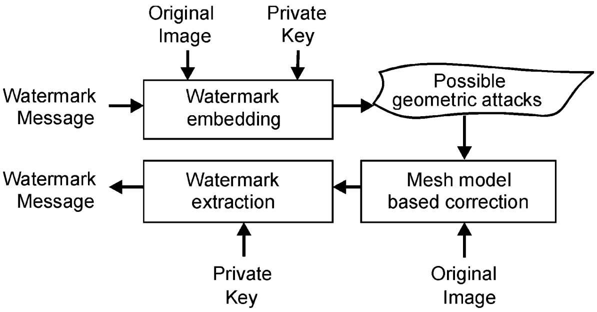


Fig. 5. Mesh model-based watermarking system.

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] and [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 into M nonoverlapping 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 nth 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 as 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:

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

where the first term is the matching error accumulated over all M mesh elements between the two images, 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 definitions for E_d: One is based on mesh regularity as in [14], which is defined as

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

where $\mathbf{t}_n = \sum_{l \in T_n}(\mathbf{p}_n - \mathbf{p}_l)$, 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 based on

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

where N the total number of mesh nodes in the image, and $\bar{\mathbf{d}}_n$ 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 as variation I and II, respectively.

In (15), ρ is a regularization parameter used to control the tradeoff 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

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

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

As an example, we show in Fig. 4(e) the corrected image from the distorted image in Fig. 4(b) using the procedure described above. As in Fig. 4(c), the difference between this corrected image and the predistortion image in Fig. 4(a) is shown in Fig. 4(f). One can see that the geometric distortion has been corrected effectively in Fig. 4(e). The regular mesh structure shown in Fig. 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 multibit watermarking, and the other on 1-bit watermarking. In the first experiment, a 50-bit watermark was embedded into a set of test images (ten 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.

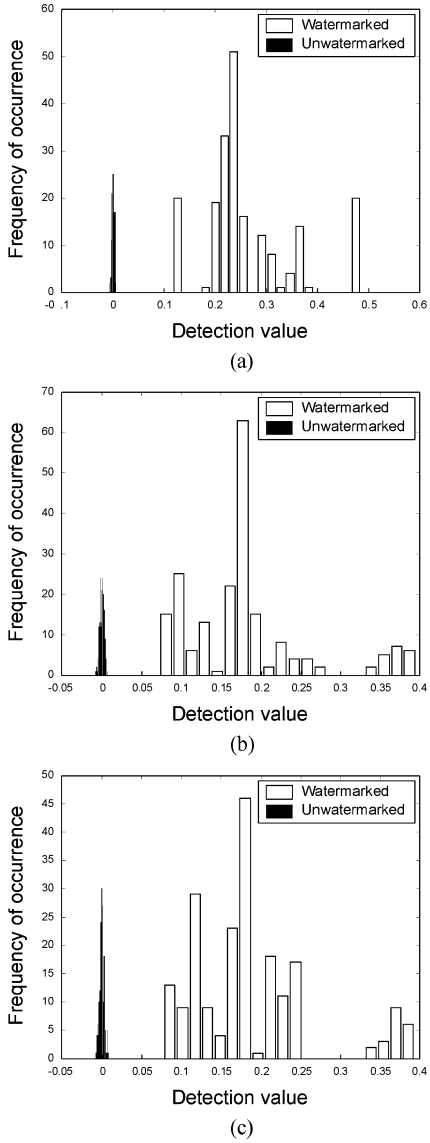
The second experiment was designed to test the proposed watermarking scheme for detection of the presence or absence of a watermark under the following geometric attacks: 1) aspect ratio changes of (1.1, 1.0), which is test case 3(c) in the distortion list given later; 2) shearing of (5%, 5%), test case 5(f) from the list; and 3) general affine transform, test case 6(a) from 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 using the three geometric attacks. The proposed algorithm was then applied to detect the presence of watermarks in these 400 images under each of the geometric attacks.

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) $-^{\circ}10$, (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 2×2, (b) median filtering 3×3, (c) median filtering 4×4, (d) sharpening by kernel $\begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}$, (e) Gaussian filtering by kernel $(1/16)\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 1 & 1 \end{pmatrix}$, 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.

TABLE I
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 proc.	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



(a)

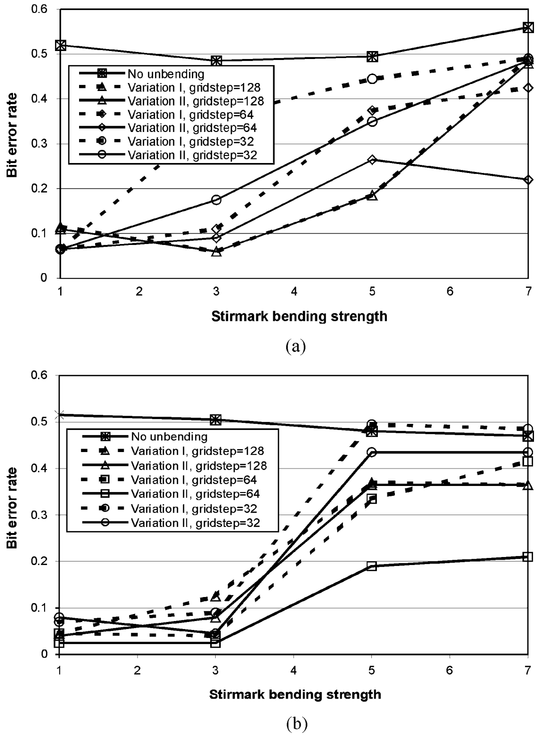
(b)

(c)

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

The test results from the first experiment are summarized in Table I. We see from these results that the proposed algorithm achieves very low decoding BER for all the geometric attacks except the 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



(a)

(b)

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

Fig. 6(a)–(c), respectively, for the three different geometric attacks used. 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.

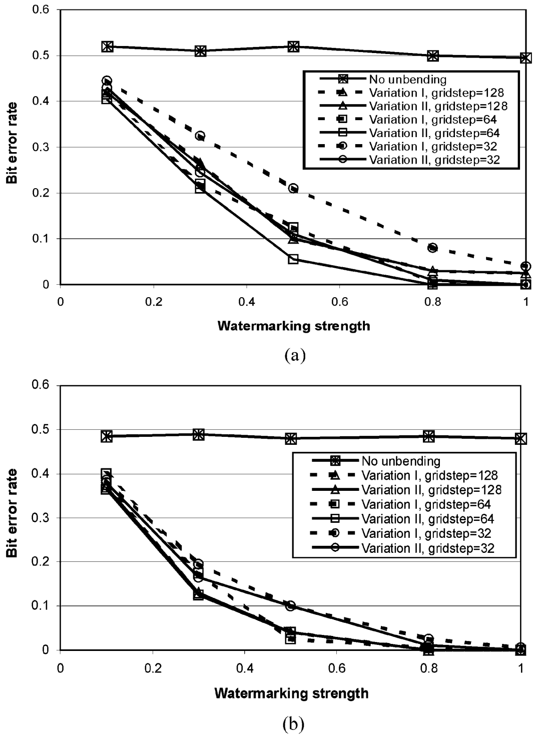
We want to mention that the proposed algorithm is not robust to overcropping, which is a common problem for the moment-based watermarking algorithms.

B. Mesh Model-Based Watermarking

For this experiment, we used Lena and Boat as test images. 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 number of experiments were performed to test the proposed watermarking system. In all experiments, the original nonwatermarked 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. 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 Versus Bending Strength: In this experiment, the watermark strength is fixed at $\lambda = 0.5$. The test results are shown in Fig. 7(a) and (b). From these results, we can see that the BER is rather insensitive to the number of mesh nodes used, especially when the bending strength is not very high.

2) BER Versus Watermarking Strength: The bending strength is fixed at 5 in this experiment, and watermarking strength λ is varied from 0.1 to 1.0. The test results are shown in Fig. 8(a) and (b). With the proposed correction, zero error decoding can be achieved when the watermarking strength λ is close to 1.0 for both images. These results indicate that the best performance was obtained with mesh elements of 64 × 64 pixels.



(a)

(b)

Fig. 8. BER versus watermark strength. (a) Lena. (b) Boat.

The minimization of (15) requires about 10 s per iteration on Pentium 4 at 1.7 GHz. This is for image size of 512 × 512 and regular mesh structure at 64 pixel nodal separation. A typical run takes about 10–20 iterations.

V. CONCLUSION

In paper, we proposed a new public watermarking algorithm that 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. By numerical experiments we demonstrate that the proposed algorithm can achieve very low decoding BER when used with multibit watermarks and perfect detection of the presence or absence of the watermark when used with single bit watermarks under various affine attacks.

We then proposed a 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 used for private watermarking where the original image is known. We tested this algorithm against random bending attacks generated by StirMark. Numerical experiments demonstrate that the proposed methodology works well.

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 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 the centering step 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 coordinates of the normalized image of $g(x,y)$ can be written as

$$\begin{pmatrix} x_n \\ y_n \end{pmatrix} = \mathbf{A}_s \mathbf{A}_y \mathbf{A}_x \begin{pmatrix} x_a \\ y_a \end{pmatrix}. \tag{A1}$$

The parameter β in the matrix $\mathbf{A}_x$ in (A1) is solved from the normalization condition in (8), i.e.,

$$\mu_{30}^{(a)} + 3\beta\mu_{21}^{(a)} + 3\beta^2\mu_{12}^{(a)} + \beta^3\mu_{03}^{(a)} = 0. \tag{A2}$$

Also, the parameter γ in the matrix $\mathbf{A}_y$ in (A1) is solved from the normalization condition in (11), i.e., it is determined as

$$\gamma = -\frac{\mu_{11}^{(2)}}{\mu_{20}^{(2)}} = -\frac{\mu_{11}^{(1)} + \beta\mu_{02}^{(1)}}{\mu_{20}^{(1)} + 2\beta\mu_{11}^{(1)} + \beta^2\mu_{02}^{(1)}}. \tag{A3}$$

A. Shearing in the x Direction

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

$$\begin{pmatrix} x_a \\ y_a \end{pmatrix} = \mathbf{A}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & \beta_0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}. \tag{A4}$$

Based on this relation, we can write the moments $\mu_{pq}^{(a)}$ in (A2) in terms of μ_{pq} using (6), and, after some algebra, we can rewrite (A2) as

$$\mu_{30} + 3(\beta+\beta_0)\mu_{21} + 3(\beta+\beta_0)^2\mu_{12} + (\beta+\beta_0)^3\mu_{03} = 0. \tag{A5}$$

Let $\beta' \triangleq \beta + \beta_0$. 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

$$\mathbf{A}_x\mathbf{A} = \begin{pmatrix} 1 & \beta \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & \beta_0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & \beta+\beta_0 \\ 0 & 1 \end{pmatrix} = \mathbf{A}'_x. \tag{A6}$$

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

B. Shearing in the y Direction

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

$$\begin{pmatrix} x_a \\ y_a \end{pmatrix} = \mathbf{A}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \gamma_0 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}. \tag{A7}$$

As above, we write the moments $\mu_{pq}^{(a)}$ in (A2) in terms of μ_{pq}, and rewrite the normalization condition as

$$\begin{aligned}\mu_{30} &+ 3\left(\frac{\beta}{1+\beta\gamma_0}\right)\mu_{21} + 3\left(\frac{\beta}{1+\beta\gamma_0}\right)^2 \\ \mu_{12} &+ \left(\frac{\beta}{1+\beta\gamma_0}\right)^3\mu_{03} = 0.\end{aligned} \tag{A8}$$

Let $\beta' \triangleq (\beta/1+\beta\gamma_0)$. 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 (A3) in terms of μ_{pq}, and obtain

$$\gamma = -\frac{\gamma_0(1+\beta\gamma_0)\mu_{20} + (1+2\beta\gamma_0)\mu_{11} + \beta\mu_{02}}{(1+\beta\gamma_0)^2\mu_{20} + 2\beta(1+\beta\gamma_0)\mu_{11} + \beta^2\mu_{02}}. \tag{A9}$$

Hence

$$\begin{aligned}\mathbf{A}_y\mathbf{A}_x\mathbf{A} &= \begin{pmatrix} 1 & 0 \\ \gamma & 1 \end{pmatrix}\begin{pmatrix} 1 & \beta \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ \gamma_0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1+\beta\gamma_0 & 0 \\ \gamma+\gamma_0(1+\beta\gamma) & 1+\beta\gamma \end{pmatrix}.\end{aligned} \tag{A10}$$

Upon some algebraic manipulation, (A10) can be rewritten as

$$\begin{aligned}\mathbf{A}_y\mathbf{A}_x\mathbf{A} = &\begin{pmatrix} 1+\beta\gamma_0 & 0 \\ 0 & \frac{1+\beta\gamma_0}{(1+\beta\gamma_0)^2\mu_{20}+2\beta(1+\beta\gamma_0)\mu_{11}+\beta^2\mu_{02}} \end{pmatrix} \\ &\times \begin{pmatrix} 1 & \beta' \\ -\mu_{11}-\beta'\mu_{02} & \mu_{20}+\beta'\mu_{11} \end{pmatrix}.\end{aligned} \tag{A11}$$

Observe the following. 1) The second matrix term in (A11) corresponds to an affine transform that is independent of the parameter γ_0, and 2) the first matrix term in (A11) corresponds to a scaling transform, which will be later absorbed into the scaling matrix $\mathbf{A}_s$ in (A1) 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_0.

C. Scaling in Both x and y Directions

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

$$\begin{pmatrix} x_a \\ y_a \end{pmatrix} = \mathbf{A}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \alpha_0 & 0 \\ 0 & \delta_0 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}. \tag{A12}$$

Again, we write the moments $\mu_{pq}^{(a)}$ in terms of μ_{pq}, and rewrite the normalization condition (A2) as

$$\mu_{30} + 3\left(\frac{\delta_0}{\alpha_0}\beta\right)\mu_{21} + 3\left(\frac{\delta_0}{\alpha_0}\beta\right)^2$$
$$\mu_{12} + \left(\frac{\delta_0}{\alpha_0}\beta\right)^3\mu_{03} = 0. \tag{A13}$$

Let $\beta' \triangleq (\delta_0/\alpha_0)\beta$. 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 (A3) in terms of μ_{pq}, and obtain

$$\gamma = -\frac{\alpha_0\delta_0\mu_{11} + \beta\delta_0^2\mu_{02}}{\alpha_0^2\mu_{20} + 2\beta\alpha_0\delta_0\mu_{11} + \beta^2\delta_0^2\mu_{02}}. \tag{A14}$$

Hence

$$\begin{aligned}\mathbf{A}_y\mathbf{A}_x\mathbf{A} &= \begin{pmatrix}1 & 0\\ \gamma & 1\end{pmatrix}\begin{pmatrix}1 & \beta\\ 0 & 1\end{pmatrix}\begin{pmatrix}\alpha_0 & 0\\ 0 & \delta_0\end{pmatrix}\\ &= \begin{pmatrix}\alpha_0 & \beta\delta_0\\ \alpha_0\gamma & \delta_0(1+\beta\gamma)\end{pmatrix}.\end{aligned} \tag{A15}$$

Upon some algebraic manipulation, (A15) can be rewritten as

$$\mathbf{A}_y\mathbf{A}_x\mathbf{A} = \begin{pmatrix}\alpha & 0\\ 0 & \frac{\alpha_0^2\delta}{\alpha_0^2\mu_{20}+2\beta\alpha_0\delta_0\mu_{11}+\beta^2\delta_0^2\mu_{02}}\end{pmatrix}$$
$$\begin{pmatrix}1 & \beta'\\ -\mu_{11}-\beta'\mu_{02} & \mu_{20}+\beta'\mu_{11}\end{pmatrix}. \tag{A16}$$

Again, the second matrix term in (A16) corresponds to an affine transform that is independent of the parameters α_0, δ_0, and the first matrix term in (A16) corresponds 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.

D. Uniqueness Under a General Affine Transforms

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

$$\mathbf{A} = \begin{pmatrix}\alpha_0 & 0\\ 0 & \delta_0\end{pmatrix}\begin{pmatrix}1 & 0\\ \gamma_0 & 1\end{pmatrix}\begin{pmatrix}1 & \beta_0\\ 0 & 1\end{pmatrix}. \tag{A17}$$

Similar to (A5), (A8), and (A13), we can derive $\beta' = (1/(\alpha_0/\delta_0\beta) + \gamma_0) + \beta_0$, where β' is a root of the normalization condition (A2) that corresponds to the original image $f(x,y)$, and β is a root corresponds to the affine transformed image. Therefore

$$\beta = \frac{\frac{\alpha_0}{\delta_0}}{\frac{1}{\beta'-\beta_0} - \gamma_0}. \tag{A18}$$

From (A18), we see that β is real if, and only if, β' is real. Thus, if (A2) 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 $|\beta'| < |(1/\gamma_0) + \beta_0|$. 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 $|\beta'| < |(1/\gamma_0) + \beta_0|$ 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 or y direction). In such a case, $|(1/\gamma_0) + \beta_0| > 4.8$. This, of course, leaves enough room for the shearing parameter β'.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers. Their comments helped improve this manuscript.

REFERENCES

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

[2] M. Kutter and F. A. P. Petitcolas, "A fair benchmark for image watermarking systems," presented at the Electronic Imaging, 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," presented at the IEEE Int. Conf. Image Processing, vol. 3, 1997.

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

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

[6] S. Pereira and T. Pun, "Robust template matching for affine resistant image watermarks," *IEEE Trans. Image Process.*, vol. 9, no. 6, pp. 1123–1129, Jun. 2000.

[7] M. Alvarez-Rodríguez and F. Pérez-González, "Analysis of pilot-based synchronization algorithms for watermarking of still images," *Signal Process.: Image Commun.*, vol. 17, no. 8, pp. 661–633, Sep. 2002.

[8] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*. San Mateo, CA: Morgan Kaufmann, 2001.

[9] P. Bas, J.-M. Chassery, and B. Macq, "Geometrically invariant watermarking using feature points," *IEEE Trans. Image Process.*, vol. 11, no. 9, pp. 1014–1028, Sep. 2002.

[10] J. Wood, "Invariant pattern recognition: A review," *Pattern Recognit.*, vol. 29, no. 1, pp. 1–17, 1996.

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

[12] StirMark 3.1 1999, F. A. P. Petitcolas. [Online]. Available: 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," presented at the Eur. Signal Processing Conf., Tampere, Finland, 2000.

[14] Y. Wang and O. Lee, "Active mesh—A feature seeking and tracking image sequence representation scheme," *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 610–624, Sep. 1994.

[15] I. Rothe, H. Susse, and K. Voss, "The method of normalization to determine invariants," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 4, pp. 366–376, Apr. 1996.

[16] D. Shen and H. S. Ip, "Generalized affine invariant image normalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, 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. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 466–476, May 1999.

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

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

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

[21] P. Dong and N. P. Galatsanos, "Affine transformation resistant watermarking based on image normalization," presented at the IEEE Int. Conf. Image Processing, Rochester, NY, Sep. 2002.

[22] P. Dong, J. Brankov, N. P. Galatsanos, and Y. Yang, "Geometric robust watermarking through mesh model based correction," presented at the IEEE Int. Conf. Image Processing, Rochester, NY, Sep. 2002.

[23] K. Aizawa and T. S. Huang, "Model-based image coding: Advanced video coding techniques for very low bit-rate applications," *Proc. 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," in *IEEE Trans. Image Process.*, vol. 6, Sep. 1997, pp. 1255–1269.

[25] A. Singh, D. Goldgof, and D. Terzopoulos, Eds., *Deformable Models in Medical Image Analysis*. Los Alamitos, CA: IEEE Computer Soc. Press, 1998.

[26] J. G. Brankov, Y. Yang, and M. N. Wernick, "4D processing of gated SPECT images using deformable mesh modeling," presented at the 6th Int. Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine, Pacific Grove, CA, 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. Image Process.*, vol. 12, no. 8, pp. 866–881, Aug. 2003.

[28] C. Podilchuk and W. Zeng, "Image adaptive watermarking using visual models," *IEEE J. Sel. Areas Commun.*, 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 Process.*, vol. 11, no. 1, pp. 16–25, Jan. 2002.

Ping Dong received the B.E. and M.E. degrees from Tsinghua University, Beijing, China, in 1988 and 1993, respectively, and the Ph.D. degree from the Illinois Institute Technology, Chicago, in 2004, all in electrical and computer engineering.

From 1988 to 1990, he was with the Tianjing Communication and Broadcasting Corporation as a Software Engineer. He was a Research Engineer with the Simulation Center, Tsinghua University, from 1990 to 1993, where he worked on projects for power plant simulators. From 1998 to 2002, he was with NMS Communication Corporation, 3com Corporation, and Motient Corporation, where he worked on various VoIP and wireless communication products. His current research interests include multimedia watermarking and digital image processing.

Jovan G. Brankov (S'99–M'02) received the Diploma of Electrical Engineering from the University of Belgrade, Belgrade, Yugoslavia, in 1996, and the M.S. and Ph.D. degrees in electrical engineering from the Illinois Institute of Technology (IIT), Chicago, in 1999 and 2002, respectively.

He is currently on the Faculty of the Electrical and Computer Engineering Department, IIT, where he is a Research Assistant Professor. His research interests include medical imaging, image sequence processing, pattern recognition, and data mining, with emphasis on biomedical applications. His current projects include the development of a multiple-image radiography modality, four-dimensional image reconstruction for SPECT, and statistical learning algorithms for image quality assessment.

Dr. Brankov has served as a technical reviewer for the IEEE TRANSACTIONS ON MEDICAL IMAGING, the IEEE TRANSACTION ON IMAGE PROCESSING, and the *Journal of Physics in Medicine and Biology*.

Nikolas P. Galatsanos (SM'95) received the Diploma of Electrical Engineering from the National Technical University of Athens, Athens, Greece, in 1982, and the M.S.E.E. and Ph.D. degrees from the Electrical and Computer Engineering Department, University of Wisconsin–Madison, in 1984 and 1989, respectively.

He was on the Faculty of the Electrical and Computer Engineering Department, Illinois Institute of Technology, Chicago, from 1989 to 2002. Currently, he is with the Department of Computer Science, University of Ioannina, Ioannina, Greece. He co-edited the book titled *Image Recovery Techniques for Image and Video Compression and Transmission* (Norwell, MA: Kluwer, 1998). His research interests center around image processing and statistical learning problems for medical imaging and visual communications applications.

Dr. Galatsanos has served as an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and the *IEEE Signal Processing Magazine*.

Yongyi Yang (M'97–SM'03) received the B.S.E.E. and M.S.E.E. degrees from Northern Jiaotong University, Beijing, China, in 1985 and 1988, respectively, and the M.S. degree in applied mathematics and the Ph.D. degree in electrical and computer engineering from the Illinois Institute of Technology (IIT), Chicago, in 1992 and 1994, respectively.

He is currently on the Faculty of the Electrical and Computer Engineering Department, IIT, where he is an Associate Professor. Prior to this position, he was a faculty member at the Institute of Information Science, Northern Jiaotong University. He is a co-author of *Vector Space Projections: A Numerical Approach to Signal and Image Processing, Neural Nets, and Optics* (New York: Wiley, 1998). His research interests are in signal and image processing, medical imaging, machine learning, pattern recognition, applied mathematical and statistical methods, real-time signal processing systems, and biomedical applications.

Franck Davoine (M'98) received the Ph.D. degree from the Institute National Polytechnique de Grenoble, Grenoble, France, in December 1995, and the best Ph.D. award from INPG, France.

After 18 months as a Postdoctorate at the University of Linkoping, Linkoping, Sweden, he joined the Compiegne University of Technology, Compiegne, France, as an Assistant Professor in 1997, and, in 2002, he joined CNRS (Heudiasyc mixed research unit), France, as a full-time Researcher. He has been or is currently involved in European and national projects concerning digital image watermarking and facial feature analysis for visual communications and human machine interaction. His research concerns image/video-based data hiding, computer vision for visual object tracking, and recognition with applications to human faces.