

Recovery of Watermarks from Distorted Images

Neil F. Johnson², Zoran Duric¹, and Sushil Jajodia²

Center for Secure Information Systems
George Mason University
Fairfax, VA 22030-4444
<http://ise.gmu.edu/~csis>
{njohnson,zduric,jajodia}@gmu.edu

Abstract. Many techniques for watermarking of digital images have appeared in numerous publications. Most of these techniques are sensitive to cropping and/or affine distortions (e.g., rotation and scaling). In this paper we describe a method for the recovery of original size and appearance of images based on the concept of *identification marks* (“fingerprints”); the method does not require the use of the “original” image, but only a small number of *salient* image points. We show that, using our method, it is possible to recover original appearances of distorted images. The restored image can be used to recover embedded watermarks.

1 Introduction

Interest in digital watermarking techniques is growing, motivated by the need to provide copyright protection to digital works, such as images, audio, and video, has increased over the last few years. Much of the interest has been driven by the growth of the Internet and the development of compression techniques and compression standards that make possible fast transmission of large volumes of information. These advances have made it easy to copy almost any song, image, video, or multimedia object that is available in digital form.

Watermarking can be used to identify owners, license information, or other information related to the digital object carrying the watermark. Watermarks may also provide mechanisms for determining if a work has been tampered with or copied illegally. In the domain of video and satellite broadcasts, watermarks are used to interfere with recording devices so copies of a broadcast are somewhat corrupt. Much of the focus has been on digital watermarking of images; this paper deals with images, although some of the discussion can be equally applied to other digital works.

Many proposed watermarking techniques are sensitive to image compression and transformations such as smoothing, rotation, scaling, cropping, and so on, or even printing and scanning. This interferes with the readability of the watermark. Publicly available tools can be used to distort images and effectively disable their

¹ Also with Computer Science Department, George Mason University.

² Also with Information and Software Engineering Department, George Mason University.

watermarks. Given the volume of data on the internet, a watermark is effectively hidden if it cannot be found using a computationally simple and fast procedure.

Attacks on watermarks may be accidental or intentional. Accidental attacks may be the result of standard image processing or compression procedures. Illicit attacks may include cryptanalysis, steganalysis, image processing techniques, or other attempts to overwrite or remove existing watermarks or confuse the reader as to the authenticity of the watermark [7, 8].

Many owners of watermarked works do not want the watermark to interfere with the use of the work by others; they therefore require that the watermark be imperceptible to the human visual system. This requirement works against the robustness of a watermark. Nevertheless, watermark users usually advertise the fact that a watermark exists.

In images, various image processing techniques and transformations are commonly employed to develop and apply digital watermarks. These methods can also be used to attack and disable watermarks. Even with advances in watermarking technology, watermarks may be forged or overwritten; for example, multiple watermarks may be placed in an image and one cannot determine which of them is valid [1]. Current watermark registration services are “first come, first serve”, and someone other than the owner of a digital work may attempt to register a copyright first.

Attacks on watermarks may not necessarily remove the watermark, but only disable its perceptibility. If watermarks are used to locate images, how can an image be located or the watermark verified after it is disabled? To begin to understand this issue, we can ask: what features of an image are unaffected by (or invariant to) the processing that disables the watermark? Finding such features is key to reliably locating an image when an embedded watermark has been disabled.

We propose alternative methods for image recognition based on the concept of *identification marks* (id-marks or “fingerprints”), which can be used to locate corrupted copies of images [3, 6]. Such a fingerprint can be used as a code to recognize an image and once an image is found, then the amount of change that occurred in the process of disabling the watermark can be determined. This same fingerprint can be used in calculating the inverse transform based on these changes over the image to recover the original appearance (scale and rotation) of the image. In doing so, we are also able to recover aspects of the embedded watermark, as we will see later.

The remainder of this paper is organized as follows: Section 2 provides formal descriptions of image processing techniques that we use in the recognition and recovery processes. Section 3 provides a brief description of our image fingerprinting and recognition methods based on invariant image properties. It shows how images can be prepared before their release to allow their easier recognition. Section 4 describes our method of recovering image parameters and appearance, and presents experimental results on images. Conclusions and suggestions about future research are presented in Section 5.

2 Affine Transforms and Displacement Fields

In this section we provide a mathematical background for our work. In Section 2.1 we formally define affine transforms and give an expression for displacement fields under affine transforms. In Section 2.2 we give the expression for normalized cross-correlation and explain its geometric meaning; we use it later in this paper to establish point correspondences in images. Finally, in Section 2.3 we introduce normal displacement fields, which we used for fine tuning of image registration parameters [3].

2.1 Affine Transforms

Let (x, y) be the image coordinates of a pixel in an image $I(x, y)$ and let the image center be at $(0, 0)$. An affine transform of $I(x, y)$ is given by

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \quad (1)$$

where (x', y') are image coordinates in the transformed image $I'(x', y')$ and $a - f$ are the transform parameters.

If we subtract the vector $(x \ y)^T$ from both sides of equation (1) we obtain an expression for the displacement $(\delta x, \delta y)$ of the point (x, y) due to the transform:

$$\begin{pmatrix} \delta x \\ \delta y \end{pmatrix} \equiv \begin{pmatrix} x' - x \\ y' - y \end{pmatrix} = \begin{pmatrix} a - 1 & b \\ c & d - 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \equiv \begin{pmatrix} a_1 & b \\ c & d_1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}. \quad (2)$$

2.2 Normalized Cross-correlation

Let $w_1 = I_1(x_1 + i, y_1 + j)$ and $w_2 = I_2(x_2 + i, y_2 + j)$, $i = -W, \dots, W$, $j = -W, \dots, W$ be two square image windows centered at locations (x_1, y_1) and (x_2, y_2) of images I_1 and I_2 , respectively. The normalized cross-correlation of w_1 and w_2 is given by

$$NCC(w_1, w_2) = \frac{(w_1 - \bar{w}_1) \cdot (w_2 - \bar{w}_2)}{\|w_1 - \bar{w}_1\| \|w_2 - \bar{w}_2\|} \quad (3)$$

where w_1 and w_2 are treated as vectors. ($a \cdot b$ stands for the inner product of vectors a and b , \bar{a} for the mean value of the vector elements and $\|a\|$ for the 2-norm of vector a .) For two windows whose pixel values differ by a scale factor only NCC will be equal to 1; if the windows are different NCC has value lower than 1. For two non-zero binary patterns which differ in all pixels NCC is -1 . Normalized cross-correlation corresponds to the cosine of the angle between w_1 and w_2 ; as this angle varies between 0° and 180° , the corresponding cosines vary between 1 and -1 .

2.3 Normal Displacement Fields

Let \mathbf{i} and \mathbf{j} be the unit vectors in the x and y directions, respectively; $\delta\mathbf{r} = \mathbf{i}\delta x + \mathbf{j}\delta y$ is the projected displacement field at the point $\mathbf{r} = x\mathbf{i} + y\mathbf{j}$. If we choose a unit direction vector $\mathbf{n}_r = n_x\mathbf{i} + n_y\mathbf{j}$ at the image point \mathbf{r} and call it the normal direction, then the *normal displacement field* at \mathbf{r} is $\delta\mathbf{r}_n = (\delta\mathbf{r} \cdot \mathbf{n}_r)\mathbf{n}_r = (n_x\delta x + n_y\delta y)\mathbf{n}_r$. \mathbf{n}_r can be chosen in various ways; the usual choice (and the one that we use) is the direction of the image intensity gradient $\mathbf{n}_r = \nabla I / \|\nabla I\|$.

Note that the normal displacement field along an edge is orthogonal to the edge direction. Thus, if at the time t we observe an edge element at position \mathbf{r} , the apparent position of that edge element at time $t + \Delta t$ will be $\mathbf{r} + \Delta t\delta\mathbf{r}_n$. This is a consequence of the well known *aperture problem*. We base our method of estimating normal displacement field on this observation.

For an image frame (say collected at time t) we find edges using an implementation of the Canny edge detector. For each edge element, say at \mathbf{r} , we resample the image locally to obtain a small window with its rows parallel to the image gradient direction $\mathbf{n}_r = \nabla I / \|\nabla I\|$. For the next image frame (collected at time $t_0 + \Delta t$) we create a larger window, typically twice as large as the maximum expected value of the magnitude of the normal displacement field. We then slide the first (smaller) window along the second (larger) window and compute the difference between the image intensities. The zero of the resulting function is at distance u_n from the origin of the second window; note that the image gradient in the second window at the positions close to u_n must be positive. Our estimate of the normal displacement field is then $-u_n$, and we call it the *normal flow*.

3 Fingerprinting Images

The task of recognizing images can be defined as matching invariant features. These features may be salient parts of images or they may be artificial additions to them. In digital watermarking the information is typically embedded into images to facilitate identification of images. The embedded information is susceptible to attack through filtering and transformations [7, 8]. To make this information robust enough it is usually necessary to distort the images to the point of making the embedded information visible.

Another approach to recognizing images is to use salient features of images as registration patterns or identification marks. In this way perceptually important image features are used for image identification. Removing these features is not possible without destroying the image. An alternative is to transform the image so that it cannot be easily recognized. Such transforms include rotating, cropping, resampling, etc. Most of these operations can be classified as affine transforms (see Section 2). They are included in widely available image processing software and are easy to perform.

In this section we introduce our methods for using salient image features in recognizing images that have been distorted by unknown affine transforms. The image features used for recognition are groups of points. [lines (edges) are also considered, but these can be considered as collections of points]. Isolated points

are not sufficient for image recognition since they are not necessarily invariant, as differing images may contain similar points. However, groups of points tend to exhibit uniqueness, For example, ratios of areas enclosed by triples of points are invariant to affine transforms [3, 6].

We have two basic approaches to our recognition process. One is based on finding strong, relatively unique points along image edges of high gradient magnitude. The other is based on finding corners with strong gradient magnitude within the image.

Regardless of which approach is taken, our image recognition method consists of two parts. First, for each image we select a set of representative feature points at multiple resolutions. Second, we use these points for recognizing images. These methods are described in [3] and [6].

3.1 Selecting Feature Points

In general, our approach is based on finding unique points in each image at multiple resolutions. The points are represented by small rectangular neighborhoods. We use neighborhood sizes ranging from 3×3 to 11×11 pixels. For each resolution we identify the unique points separately. The selected points usually differ when resolution changes. In this section, we will summarize our two techniques for identifying image feature points used to construct an “image fingerprint.” The first approach we will discuss is based on finding relatively unique points along edges within an image. The second method is a further refinement of the first approach by considering corners of edges and less emphasis on uniqueness of the individual points.

Edge-based Feature Points This method of choosing unique feature points consists of several steps. First, we compute the image gradient ∇I over the image. We identify the image points that have large values of the gradient magnitude $\|\nabla I\|$. Note that these points typically correspond to edges (see Figure 1). We consider all the points that have gradient magnitude larger than one third of the highest gradient magnitude in the entire image. In doing so we insure that the second selection step operates on a smaller number of image points.

Second, for each of the selected points (x_i, y_i) we compute the similarity of its neighborhood, centered at (p, q) , to the neighborhoods of other points in the image. In the remainder of this paper we will use the term *image point* to represent the point and its neighborhood. We use the normalized cross-correlation (see Section 2.2) as the similarity measure. For an image point (p, q) we obtain the similarity function $s_{p,q}(x - p, y - q)$. This function has a local maximum at $s_{p,q}(0, 0) = 1$ since the value at $(0, 0)$ corresponds to the similarity of the point with itself. If the point is *unique*, i.e. there are no other points in the image that are similar to it, $s_{p,q}(0, 0)$ is the global maximum of $s_{p,q}$ as well. If the point is unique we consider the sharpness of the peak at $(0, 0)$ and the next highest value of $s_{p,q}$ to decide if the point is a feature point.

Figure 2 shows three examples of feature point selection. The points on the left and right are good feature points; their similarity functions computed over a



Fig. 1. Identifying likely feature points: (a) an original image and (b) the image points with large values of gradient magnitude.

60×60 pixel window (lower row) have sharp peaks at the center (cross-correlation with itself), while all other similarity values are below 0.5. The center point is not a good feature point; it can be seen that its similarity function (the middle of the lower row of Figure 2) does not have a sharp peak at the center, and there are multiple other points with similarity values around 0.8 in the 60×60 pixels window centered at the point.

Corner-based Feature Points We rely on the cornerness measure for salient point selection. Consider the spatial image gradient $[E_x, E_y]^T$, computed for all points (x, y) of an image area (neighborhood) A . The matrix M is defined as

$$M = \begin{pmatrix} E_x^2 & E_x E_y \\ E_x E_y & E_y^2 \end{pmatrix} \quad (4)$$

where the sums are taken over the image neighborhood captures the geometric structure of the gray level pattern of A . M is a symmetric matrix and can therefore be diagonalized by rotation of the coordinate axes, with no loss of generality, we can think of M as a diagonal matrix [4]:

$$M = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \quad (5)$$

where λ_1 and λ_2 are eigenvalues of M . We can choose λ_1 as the larger eigenvalue so that $\lambda_1 \geq \lambda_2 \geq 0$. If A contains a corner, then we expect $\lambda_1 \geq \lambda_2 > 0$, and the larger the eigenvalues, the stronger (higher contrast) their corresponding edges. A corner is identified as two strong edges; therefore as $\lambda_1 \geq \lambda_2$, a corner is a location where λ_2 is sufficiently large [10].

Given this The method of choosing feature points based on corners consists of several steps. First, we identify the corner points that have cornerness measure larger than some threshold. We consider all the points that have cornerness

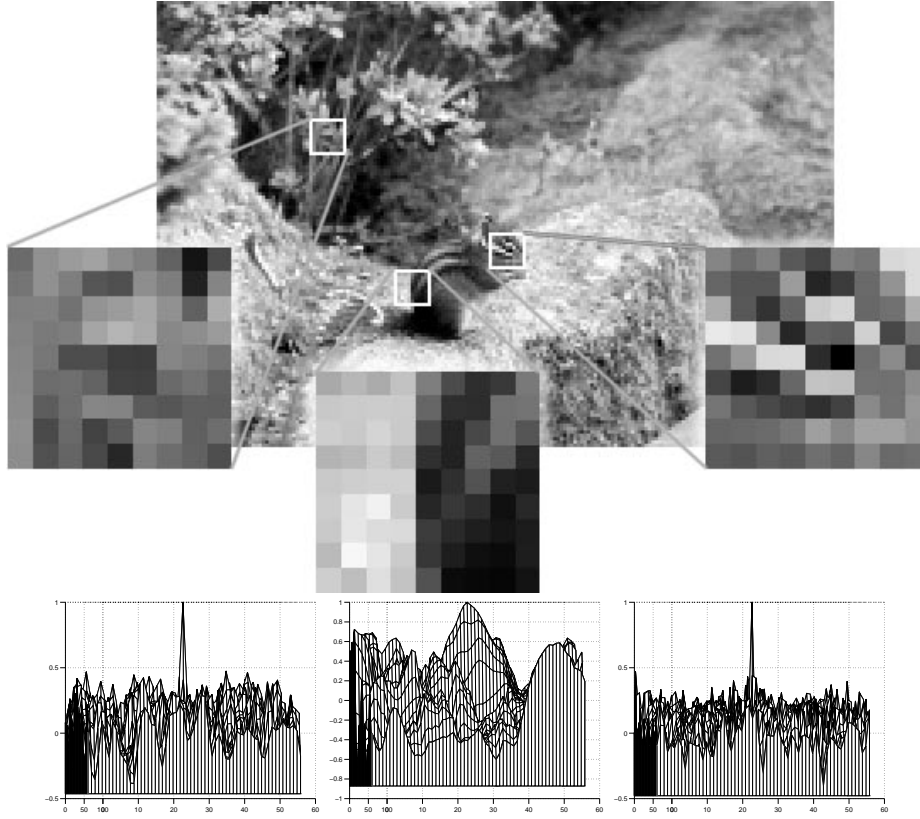


Fig. 2. Selecting feature points. Upper row: 9×9 neighborhoods of three image points. Lower row: corresponding similarity functions shown from an 80° viewing angle. The two outside points are good feature point candidates, while the center point is not.

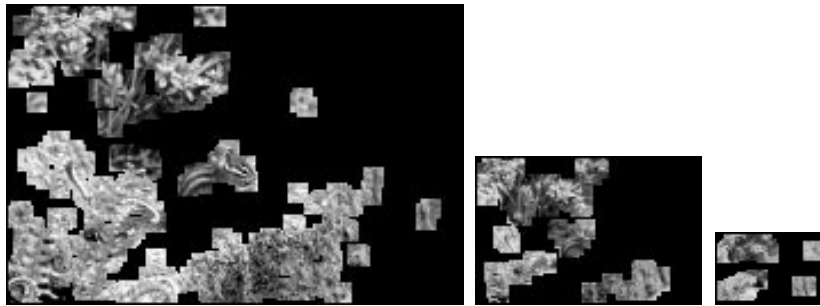


Fig. 3. Selected feature points (based on *edges*) for $1/2$, $1/4$, and $1/8$ resolutions; each point is shown with its 9×9 neighborhood.

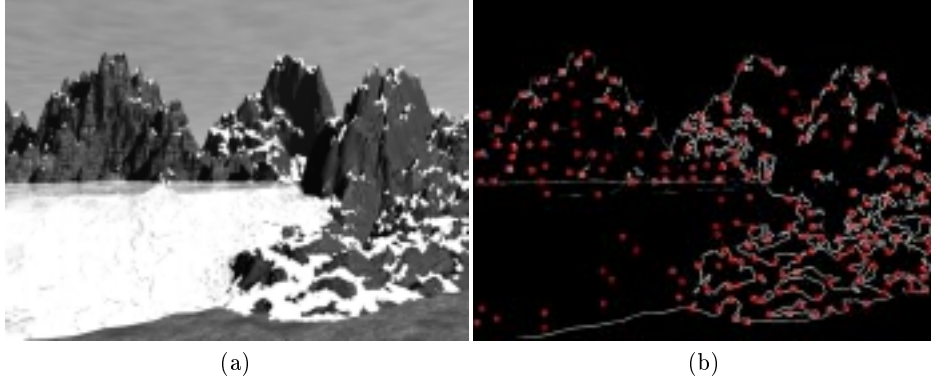


Fig. 4. Identifying likely feature points: (a) an original image. (b) The image points with large values of gradient magnitude and image points (in red) with strong cornerness measures.

measure larger than one-twentieth of the highest cornerness measure in the entire image (see Figure 4). Also, values for λ_2 must be significantly large and $\lambda_1 < 2.5\lambda_2$.

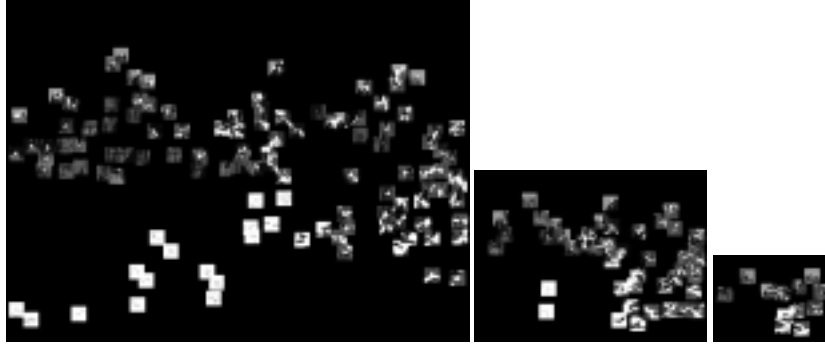


Fig. 5. Selected feature *corner* points for full, 1/2, and 1/4 resolutions; each point is shown with its neighborhood.

Second, for each of the selected corner points we compute the similarity of its neighborhood to the neighborhoods of other corner points in the image. In the remainder of this paper we will use the term image point and feature point to represent the point and its neighborhood. We use the normalized cross-correlation as the similarity measure (see [3] for details). This process is applied at multiple resolutions. Figure 5 shows selected feature points at three different resolutions. Notice that the number of feature points goes down as the resolution decreases.

Also, image points that are not unique at higher resolutions may become prominent at lower resolutions. This is to counter scaling and blurring so the image can still be recognized. These sets of feature points provide a compressed representation of the original image and can be used to identify variations of the same image.

The processes described above for feature point selection are applied at multiple resolutions. Typically, as resolution changes selected feature points also change. Figure 3 shows selected feature points for the edge-based method at three different resolutions. Figure 5 shows selected feature points for the corner-based method at three different resolutions. Both of these figures illustrate that the number of feature points goes down rapidly as the resolution decreases. Also, image points that are not unique at higher resolutions may become prominent at lower resolutions.

4 Recovering Watermarks

In this section we describe our method of recovering the original size and aspect of a distorted image given the feature points identified in Section 3.1. Following the recovery process watermarks that may have been embedded in the image can be retrieved.

4.1 Estimating Transform Parameters

In this section we describe how the image size and aspect can be recovered by using the correspondences between image points in the original (I) and transformed (I') images.

Let (x_i, y_i) , $i = 1, \dots, N$ be image points in the image I and let (x'_i, y'_i) , $i = 1, \dots, N$ be the corresponding points in the image I' , respectively. From (1) we have

$$\begin{pmatrix} x'_i \\ y'_i \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}, \quad i = 1, \dots, N. \quad (6)$$

We can rewrite equations (6) as

$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ \vdots & & & & & \\ x_N & y_N & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ \vdots & & & & & \\ 0 & 0 & 0 & x_N & y_N & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ e \\ c \\ d \\ f \end{pmatrix} = \begin{pmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_N \\ y'_1 \\ y'_2 \\ \vdots \\ y'_N \end{pmatrix}. \quad (7)$$

Equation (7) can be written as

$$\mathbf{A}\mathbf{u} = \mathbf{b} \quad (8)$$

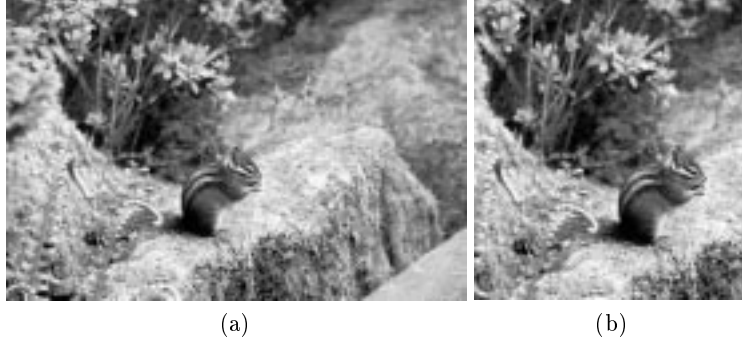


Fig. 6. Recognizing images: (a) A distorted version of the image from Figure 1a. (b) A cropped version of image (a).

where \mathbf{A} , \mathbf{u} , and \mathbf{b} are defined by comparing equations (7) and (8).

We seek \mathbf{u} that minimizes $\|E\| = \|\mathbf{b} - \mathbf{A}\mathbf{u}\|$; the solution satisfies the system [9]

$$\mathbf{A}^T \mathbf{A} \mathbf{u} = \mathbf{A}^T \mathbf{b} = \mathbf{d}. \quad (9)$$

We observe that the problem can be further simplified if we rewrite (8) as

$$\begin{pmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_1 \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix} \quad (10)$$

where \mathbf{A}_1 , \mathbf{u}_1 , \mathbf{u}_2 , \mathbf{b}_1 , and \mathbf{b}_2 are defined by comparing equations (7) and (10). Equation (9) thus separates into two equations:

$$\mathbf{A}_1^T \mathbf{A}_1 \mathbf{u}_1 = \mathbf{A}_1^T \mathbf{b}_1, \quad \mathbf{A}_1^T \mathbf{A}_1 \mathbf{u}_2 = \mathbf{A}_1^T \mathbf{b}_2 \quad (11)$$

We solve these systems using the Cholesky decomposition [9]. Since the matrix $\mathbf{A}_1^T \mathbf{A}_1$ is a positive definite 3×3 matrix there exists a lower triangular matrix L such that $\mathbf{L}\mathbf{L}^T = \mathbf{A}_1^T \mathbf{A}_1$. We solve two triangular systems $\mathbf{L}\mathbf{e}_1 = \mathbf{d}_1 = \mathbf{A}_1^T \mathbf{b}_1$ and $\mathbf{L}^T \mathbf{u}_1 = \mathbf{e}_1$ for \mathbf{u}_1 and similarly for \mathbf{u}_2 . Note that we need only one decomposition for both systems.

The computed \mathbf{u} may be inaccurate due to various geometrical and numerical factors, to be discussed below. Given the estimate \mathbf{u} , based on point correspondences between images I and I' , we use equation (6) to obtain the inverse affine transform of I' ; we call this corrected image $I^{(1)}$. [The inversion of (6) is obtained implicitly. For each pixel position (x, y) of $I^{(1)}$ we compute the pixel position (x', y') in I' (note that x' and y' may be non-integers). We obtain the gray level for (x, y) by interpolating the gray levels of transformed image I' .]

Figure 6 shows images derived from the “original” images in Figure 1a and Figure 4. Figure 6a was created by applying an affine transform (cropping, scaling, and rotating) to the original. Figure 6b was created by cropping the image in Figure 6a.



Fig. 7. Recovering image size and aspect: (a) The recovered image from Figure 6a. (b) The recovered image from Figure 6b.

Figure 7 shows results of recovering the size and aspect of a distorted image using our method for the images in Figure 6. The estimated affine transform parameters $a - f$ for the image in Figure 6a were $(1.082 \ 0.004 \ -0.015 \ 1.015 \ -8.51 \ 2.32)^T$; the corresponding inverse transform applied to the image resulted in the image shown in Figure 7a. Similarly, the estimated affine transform parameters for the image in Figure 6b were $(1.084 \ 0.004 \ -0.014 \ 1.015 \ -20.85 \ -34.81)^T$; the corresponding inverse transform applied to the image resulted in the image shown in Figure 7b. Note that the recovered parameters are very similar for both the uncropped and cropped images. Note also that the embedded watermark [2] has been successfully detected in the recovered images.

In this section we discussed various factors that may contribute to inaccuracies in estimating $\mathbf{u} = (a \ b \ e \ c \ d \ f)^T$. However, it is possible to iteratively improve on the computed solution of the system (11) using the normal flow. The technique for the refinement of image recovery based on normal displacement is explained in [3]. We show experimental results in Section 4.2.

4.2 Additional Experiments

We have experimented with the images shown in previous sections and with the images shown in Figure 8 using both a commercially available watermarking tool [2] and the watermarking technique described in [5]. A demo of the commercial watermark is available with Adobe Photoshop. The technique described in [5] embeds a watermark that corresponds to a logo or a text image into the original/cover image. We have successfully recovered the watermark in all cases. In some instances we had to go through the refinement phase to recover the watermark. An example of watermark recovery is shown in Figure 9.

Figure 9 shows an example of a mask-based watermark and recovery after attack. The image is watermarked using a mask to produce the watermarked image (see Figure 9a). The watermark is not visible, but the enhanced image difference reveals it (see Figure 9b). An attack on the watermark is conducted



Fig. 8. Additional images used in our experiments.

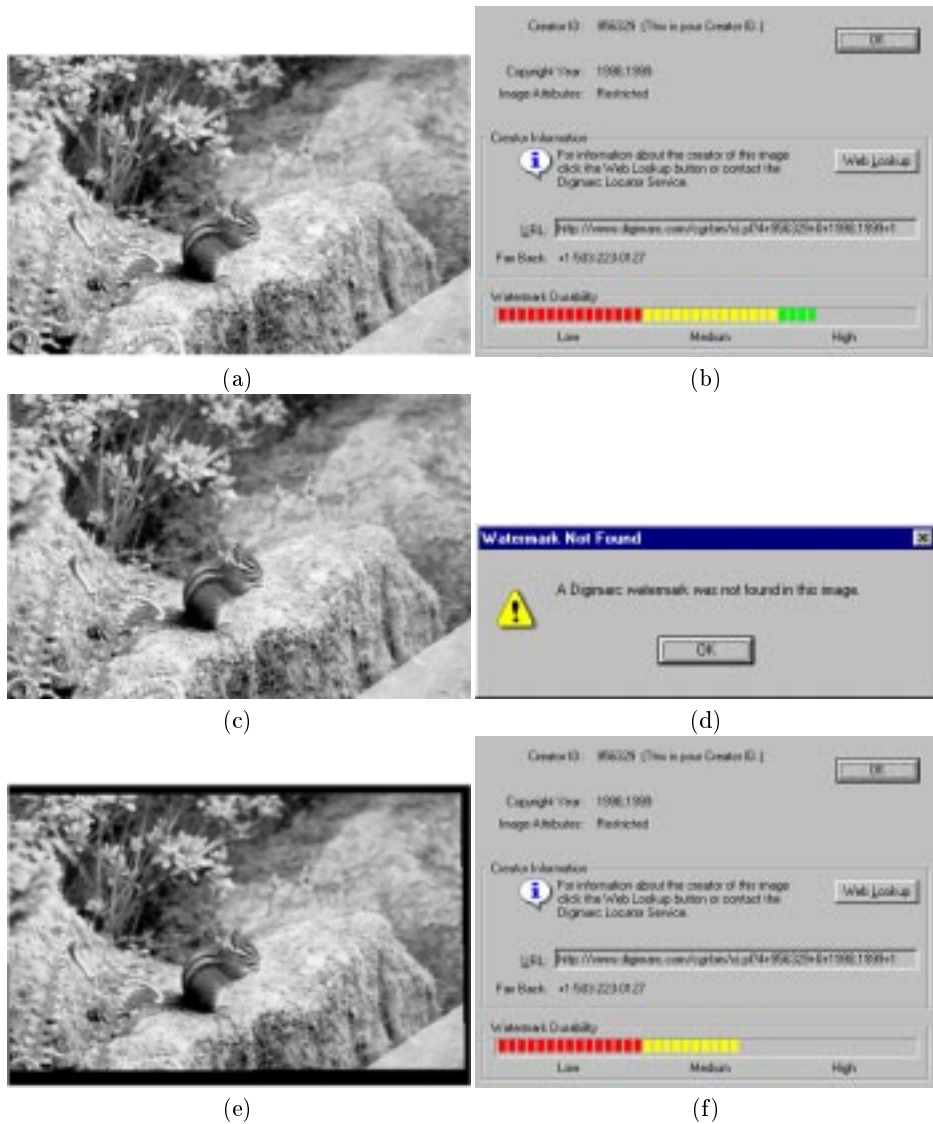


Fig. 9. Recovery of a commercial watermark [2] for an image in Figure 8. (a) The watermarked image. (b) Detection of the commercial watermark embedded in (a). (c) The distorted image. (d) Error message when attempting to read the watermark from (c). (e) The recovered image. (f) The watermark is once again available and can be read from (e).

by applying Stirmark against the watermarked image (see Figure 9c). Figure 9d shows the enhanced difference between the original image and the distorted image; the watermark is not visible. The affine transformation parameters were estimated as $(1.0255 \ 0.0012 \ -0.0049 \ 1.0045 \ 0.9685 \ 1.0939)^T$. The recovered image is shown in Figure 9e. Finally, Figure 9f shows the enhanced difference between the recovered image and the original image, revealing the watermark.

Figure 10 shows the results of refining the size and aspect of a distorted image using our method. [This method is applied when the distortion of an image is small; this typically happens after applying the recovery method described in the previous section.] Figure 10a shows the original image. Figure 10b shows the normal displacement field between the original image and the distorted image (not shown). The affine transform parameters estimated from the normal displacement field using the method described in this section are $(0.9993 \ -0.0002 \ -0.0029 \ 1.0002 \ -0.7565 \ -0.842)^T$. Figure 10c shows the recovered image that was obtained by applying the inverse affine transform to the distorted image. Finally, Figure 10d shows the normal displacement field between the images in Figure 10a and Figure 10c. The affine transform parameters estimated from this normal displacement field are $(1 \ -0.0002 \ 0.0001 \ 1.0001 \ 0.0356 \ -0.0206)^T$. Since the transform is small (the induced normal displacement field is < 0.5 everywhere) no further refinement is needed.

5 Conclusion and Future Directions

Digital works are subject to illicit copying and distribution. Many owners wish for some means of identifying ownership and copyright. Digital watermarks can fulfill this role and provide a means to identify and track digital works. However, embedded watermarks may fail to be recognized due to accidental corruption or attack by cropping and/or affine distortions (e.g., rotation, scaling, and blurring) [7, 8]. This hampers the ability to locate and identify watermarked images over distributed networks such as the Internet.

In this paper, we introduced alternative methods for image recovery, based on inherent features within images that can be used to "fingerprint" images. These identification marks can be applied to locate images and recover image size and aspect from distorted images. Our methods do not rely on embedded information and can be used to recover images distorted by various geometric transformations [3, 6].

We provided examples showing that it is possible to recover the original appearances of distorted images. In many cases doing so results in the recovery of embedded watermarks. We continue to investigate ways to make these methods more efficient for image recognition and recovery.

References

1. Craver, S., Memon, N., Yeo, B., Yeung, N.M., "Resolving Rightful Ownerships with Invisible Watermarking Techniques: Limitations, Attacks, and Implications," *IEEE Journal on Selected Areas in Communications*, 16(4):573–586, 1998.

2. Digimarc Corporation, PictureMarcTM, MarcSpiderTM, <http://www.digimarc.com>
3. Duric, Z., Johnson, N.F., and Jajodia, S., "Recovering Watermarks from Images," submitted to *IEEE Transactions on Image Processing*, 1999.
4. Jahne, B.: *Digital Image Processing*, 4th ed. Springer-Verlag, 1997.
5. Johnson, N.F., Duric, Z., and Jajodia, S., "A Role for Digital Watermarking in Electronic Commerce," *ACM Computing Surveys*, 1999.
6. Johnson, N.F., Duric, Z., and Jajodia, S., "On "Fingerprinting" Images for Recognition," to appear in Fifth International Workshop on Multimedia Information Systems (MIS'99), Indian Wells, California, 1999.
7. Johnson, N.F., Jajodia, S., "Steganalysis of Images Created using Current Steganography Software," in Aucsmith, D., (ed.), *Second Information Hiding Workshop*, Portland, Oregon, *Lecture Notes in Computer Science*, Vol. 1525. Springer-Verlag, 1998.
8. Petitcolas, F., Anderson, R., Kuhn, M., "Attacks on Copyright Marking Systems," in Aucsmith, D., (ed.), *Second Information Hiding Workshop*, Portland, Oregon, *Lecture Notes in Computer Science*, Vol. 1525. Springer-Verlag, 1998.
9. Stewart, G.W. *Introduction to Matrix Computations*. Academic Press, New York, 1973.
10. Trucco, E. and Verri, A.: *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.

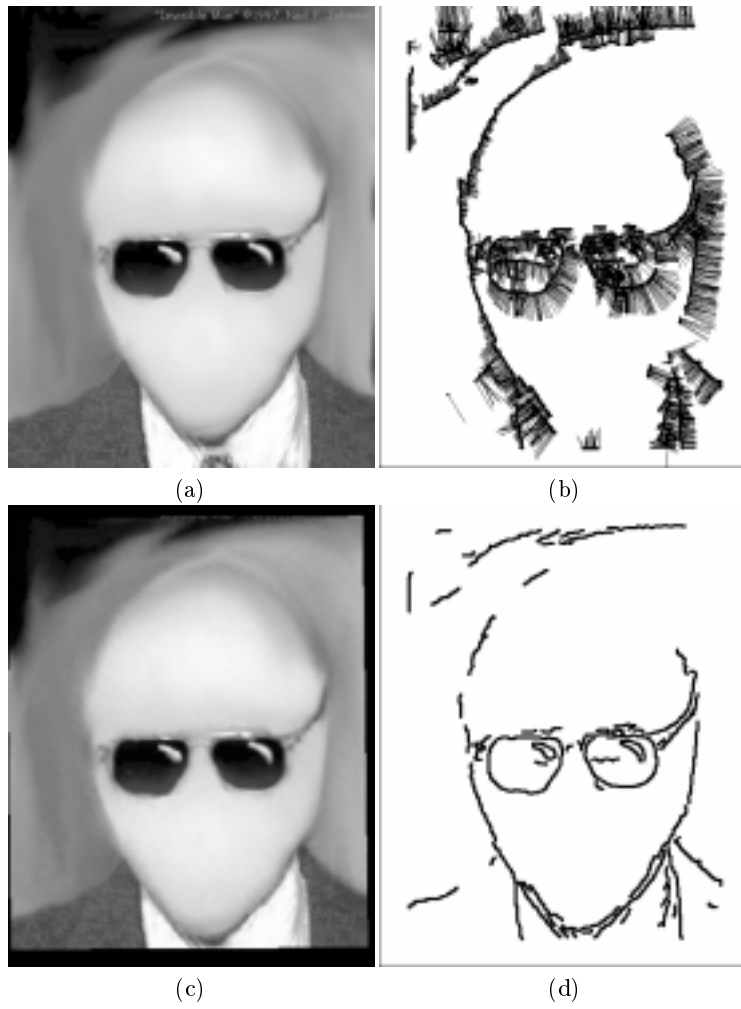


Fig. 10. Refining image size and aspect: (a) The original image. (c) The normal displacement between the original and the distorted image (not shown). (c) The recovered image. (d) The normal displacement field between images (a) and (c).