Recovering Watermarks from Images

Zoran Duric[1], Neil F. Johnson[2], Sushil Jajodia[2]

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

April 15, 1999

Abstract

Many techniques for watermarking of digital images have appeared recently. 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 recognizing images based on the concept of *identification marks*; 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 recognize distorted images and recover their original appearances. Once the image is recognized we use a second technique based on *normal flow* to fine-tune image parameters. The restored image can be used to recover watermarks that were embedded in the images by their owner.

---

[1] Also with Computer Science Department, George Mason University.
[2] Also with Information and Software Engineering Department, George Mason University.

1 Introduction

Recently, there has been much interest in watermarking techniques for digital works, such as images, audio, and video. 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.

Interest in digital watermarks is growing, motivated by the need to provide copyright protection to digital works. A number of hardware and software technologies are being developed to deter illicit copying. 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 ingenious watermarking methods have been proposed and implemented. However, some of the goals of watermarking have not been fulfilled. Many proposed techniques are sensitive to image compression and transformations such as smoothing, rotation, scaling, cropping, and so on, or even printing and scanning. This allows watermarks to be suppressed by making small changes to the images. Indeed, there are publicly available tools that can be used to distort images and effectivelly hide their 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. Some authors have questioned the belief that watermarking can be used to unambiguously prove ownership. They maintain that the legal obstacles to using digital watermarks have not yet been resolved and that these obstacles limit the usefulness of watermarking. This has sparked a debate in the security community as to whether it is necessary to register all digital works with trusted third parties before their release so that ownership can be proven, should an illicit copy of the work be found.

Based on these observations, we revisit the goals of digital watermarking and propose that finding illicit copies of images should be the first goal. In other words, we are interested in recognizing images effectively despite changes made to hide the fact that they have been copied. To this end we define the concept of an *identification mark* (id-mark), which can be used to make it easier to find corrupted copies of images. An id-mark can be used as a code to recognize an image and/or as a registration pattern that can be used to recover the original appearance (scale and rotation) of the image. Images that are recognized in this manner may be compared to stored originals to further refine their scale and orientation and *align* them with their respective originals; we show

how *normal flow* can be used to accomplish this refinement. Once an image is found and aligned with its original we may be interested in discovering from which image the copy was made. We leave the issue of legal proof of original ownership to the computer security and legal communities.

The remainder of this paper is organized as follows: Section 2 provides an introduction to digital watermarking and a brief survey of related research. This section introduces and informally describes some of the most common attacks on watermarks. Formal descriptions of image processing techniques that can be used to disable and/or recover watermarks are provided in Section 3. Section 4 describes our method of recognizing distorted images based on invariant image properties. It shows how images can be prepared before their release to allow their easier recognition. Section 5 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 6.

2   Background and Related Work

2.1   Digital Watermarking

Applications of digital watermarking range from identifying ownership and copyright to providing tracking capabilities for digital media. Watermarking techniques can be classified in several ways: as perceptible or imperceptible; as applied in the space or in a transform domain; and as to how the watermark is recovered or revealed (specifically: is an original required or not?).

A method of watermarking digital images should have several characteristics. The watermark should be integrated with the image content so it cannot be removed easily without severely degrading the image. It should require no additional image formatting or storage space, and it should not degrade the image to a degree that interferes with its usefulness. (The visible watermarks (logos) in many television broadcasts are considered eyesores by some, while others simply ignore them. What constitutes interference with an image is subjective and depends on the end user.) Various sorts of information can be stored in a watermark, including license, copyright, copy control, content authentication, and tracking information. A watermark should be fairly tamper-resistant and robust to common signal distortions, compression, and malicious attempts to remove the watermark.

Perceptible or Imperceptible
Digital watermarks may be perceptible or imperceptible. Imperceptible watermarks cannot be detected by the human senses, but can be read by a computer. Many authors feel that image-based digital watermarks should be invisible to the human eye. If the watermark is supposed to be imperceptible, there is a debate as to whether the existence of the watermark should be advertised. Advertising the presence of watermarks invites hackers to attempt to alter or disable them. If media can be manipulated by legitimate means to embed a watermark, illicit information can

also be placed in the same imperceptible space [JJ98b]. Other authors prefer visible watermarks, and clearly advertise the existence of watermarks, to deter illicit handling or theft of the images. Both viewpoints have merit; but the determination must be made by the owner of the images and depends on the intended use of the watermarked work. Some watermarking techniques can be used to determine if there has been any tampering with the work; other watermarking methods may be used to track works to and from licensed users.

Spatial or Transform Domain
Another classification depends on whether the watermark is applied in the space domain or in a transform domain. Tools used in the space domain include bit-wise techniques such as least significant bit (LSB) or *noise* insertion and manipulation [CKSL96]. Patterns placed in the image [Car95] and spatial relationships between image components are other *additive* forms of watermarking. Techniques that provide additive information such as masking techniques [JJ98] without applying a function of the image to determine the watermark location are also categorized as being in the space domain, though they share the survivability properties of transform domain techniques.

The transform domain class of watermarks includes those that manipulate image transforms. Early work in this area considered the possibility that dithering process used for image quantization might be used to hide information [TNM90]. Transforms such as the fast Fourier transform (FFT), discrete cosine transform (DCT) [KRZ94, KZ95, ODB96], and wavelet transform [KH97, XBA97] hide information in the transform coefficients. Many variations on these approaches exist, ranging from applying the transform to the entire image [CKLS95, HW96] to applying it to blocks of the image [Dig, Sig, SZT96], or applying methods similar to those used in JPEG image compression [GW92, BS95]. These methods hide messages in relatively significant areas of the cover and may manipulate image properties such as luminance. Transform domain watermarking and masking techniques are more robust to attacks such as compression, cropping, and image processing techniques in which significant bits are changed.

Both space domain and transform domain methods may employ patchwork, pattern block encoding, or spread spectrum concepts which may add redundancy to the hidden information [BGML96, CKLS95, SC96]. These approaches help protect against some types of image processing such as cropping and rotating. The patchwork approach uses a pseudo-random technique to select multiple areas (or patches) of an image for marking [Rho97]. Each patch may contain the watermark, so if one is destroyed or cropped, the others may survive. The message data becomes an integral part of the image by adjusting its luminance values, as in masking [JJ98].

Original Required?
The key to any watermarking method is the ability to read the embedded watermark. Bit-wise or noise dependent methods read the watermark without needing an original for comparison. However, these methods are vulnerable to small changes in the images and they yield relatively weak

watermarks. Transform domain watermarking techniques that do not require using the original to extract the watermark are presented in [PBBC97, FH97, OP97]. Methods that do not require the original permit faster recognition of the embedded data (since time is not needed to obtain the original image); have larger information capacity ("payload"); do not require use of a third party for media recognition or registration; and can be applied by search engines to locate images over networks. However, such methods are typically more fragile and can be disabled with little processing.

Space domain methods such as masking, and many transform-domain watermarking techniques, depend on storing an original image for comparison to read the watermarks [CKLS95, KH97, ODB96, XBA97]. Typically, these methods are made more robust to tampering by applying redundancy and masking techniques. They use a registration service or some other method that maintains a *clean original*. This may slow down the watermark reading process, but provides a layer of authentication if a third-party registration service is used (such as the U.S. Library of Congress for copyrighted material).

A tradeoff exists between a watermark's payload and its robustness to manipulation. Watermarks typically hide very little information and rely in part on redundancy of the mark to survive attacks such as cropping. This approach has a low bandwidth for passing hidden information. Bit-wise methods typically have the capacity to hide larger amounts of information in a cover; but these methods are vulnerable to attacks such as cropping. If the embedding method relies on the noise level (LSB) of the cover, little processing is required to disable reading the embedded message. This may be desirable if the purpose of the embedded information is to determine whether the medium has been altered. If the watermark can be retrieved, then the medium has not been altered. However, loss of the watermark results when even small changes occur in the medium.

2.2   Attacks on Watermarks

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 [JJ98b, PAK98].

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.

Compression - Noise Reduction
Compressing the carrier of the watermark may inadvertently render the watermark useless. This

is especially likely for the bit-wise image domain methods used for watermarking. The original message can be reconstructed exactly if a lossless compression method is used. Lossy compression methods may yield better compression, but may not maintain the integrity of the original image.

Cryptanalysis

Some methods require a password to detect a watermark; these methods are therefore vulnerable to cryptanalysis such as brute-force or dictionary attacks. Historically, the term *dictionary attack* refers to finding passwords by checking a list of terms. With improved processor speeds, a brute-force approach can find passwords by exhaustive search instead of using a dictionary list. Brute-force and dictionary attacks are general threats to passwords. Since the passwords used in watermarking are typically small by cryptographic standards, guessing character combinations until the correct guess is made can often identify them.

Image Processing and Transformations

Image processing 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 [CMYY98]. 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. Some watermarking tools are distributed with over-the-shelf software.

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. Once an image is found, then the amount of change that occurred in the process of disabling the watermark can be determined. Applying an inverse transform based on these changes over the image will recover aspects of the embedded watermark, as we will see later.

3   Affine Transforms and Displacement Fields

In this section we provide mathematical preliminaries for our work. In Section 3.1 we formally define affine transforms and give an expression for displacement fields under affine transforms. In Section 3.2 we describe some image properties that remain invariant under affine transforms and thus can be used for image recognition. In Section 3.3 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 3.4 we introduce normal displacement fields, which

will be used in this paper for fine tuning of image registration parameters.

3.1   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

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

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:

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

3.2   Affine Invariants

Given two images I and I', such that I' can be obtained through an affine transform of I, we are interested in features of I that remain unchanged in I'; these features are usually called *geometric invariants* [Wei93]. Let $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$, and $P_3 = (x_3, y_3)$ be three noncollinear image points in the image I and let $P'_1 = (x_1, y_1)$, $P'_2 = (x_2, y_2)$, and $P'_3 = (x_3, y_3)$ be their corresponding points in the image I'. The mapping between the points of I and I' is given by (1). The area of the triangle $\triangle P_1 P_2 P_3$ is given by the determinant

$$S_{123} = \frac{1}{2} \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \tag{3}$$

and the area of the corresponding triangle $\triangle P'_1 P'_2 P'_3$ is given by

$$S'_{123} = (ad - bc) S_{123} \tag{4}$$

where a, b, c, and d are given by (1). The area of the triangle formed by three noncollinear points is a *relative affine invariant* of image I [Wei93]. Since $(ad - bc)$ in (4) does not change for triples of image points, the ratio of the areas of two triangles is an *absolute affine invariant*.

3.3 Normalized Cross-correlation

Let $w_1 = I_1(x_1 + i, y_1 + j)$ and $w_2 = I_2(x_2 + i, y_2 + j)$, $i = -W, \ldots, W$, $j = -W, \ldots, W$ 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

$$NCC(w_1, w_2) = \frac{(w_1 - \overline{w}_1) \cdot (w_2 - \overline{w}_2)}{\|w_1 - \overline{w}_1\| \, \|w_2 - \overline{w}_2\|} \tag{5}$$

where w_1 and w_2 are treated as vectors. ($a \cdot b$ stands for the inner product of vectors a and b, $\overline{a}$ 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.

3.4 Normal Displacement Fields

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

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 $\vec{r}$, the apparent position of that edge element at time $t + \Delta t$ will be $\vec{r} + \Delta t \delta\vec{r}_n$. 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 $\vec{r}$, we resample the image locally to obtain a small window with its rows parallel to the image gradient direction $\vec{n}_r = \nabla I / \|\nabla I\|$. 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*.

4 Recognizing Distorted 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 [JJ98b, PAK98]. 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 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 3). They are included in widely available image processing software and are easy to perform.

In this section we describe our method of recognizing images that have been subjected to unknown affine transforms, using salient image features. The image features that are typically used for recognition include points and lines. Points are more general since lines can be defined using collections of points. Typically 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 exibit uniqueness, For example, ratios of areas enclosed by triples of points are invariant to affine transforms (see Section 3.2).

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. The method is described below.

4.1 Selecting Feature Points

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 from 5×5 to 11×11 pixels. For each resolution we identify the unique points separately. The selected points usually differ when resolution changes.

Our 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

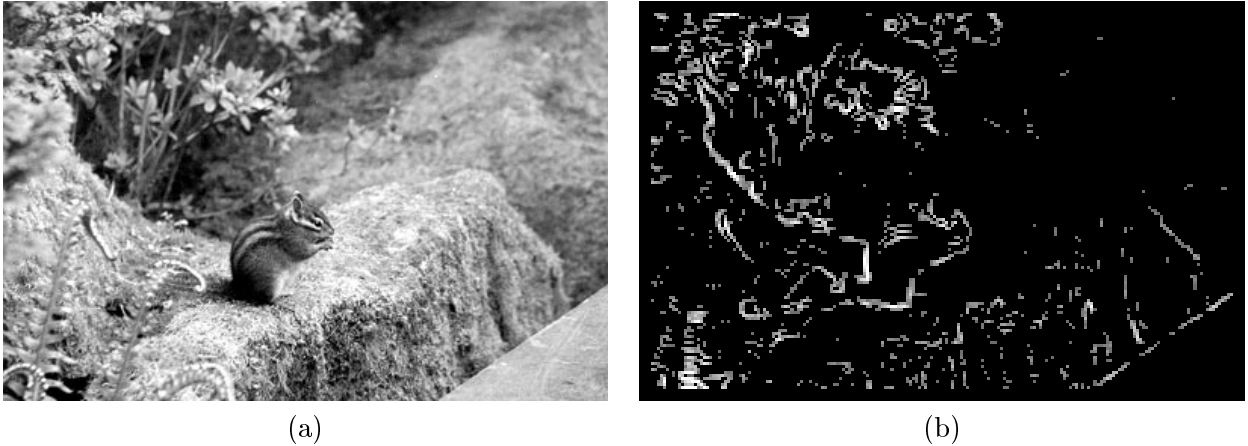<div align="center">(a)   (b)</div>

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

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[1], 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 3.3) 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 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.

The process described above is applied at multiple resolutions. Typically, as resolution changes selected feature points also change. Figure 3 shows selected feature points at three different res-

---

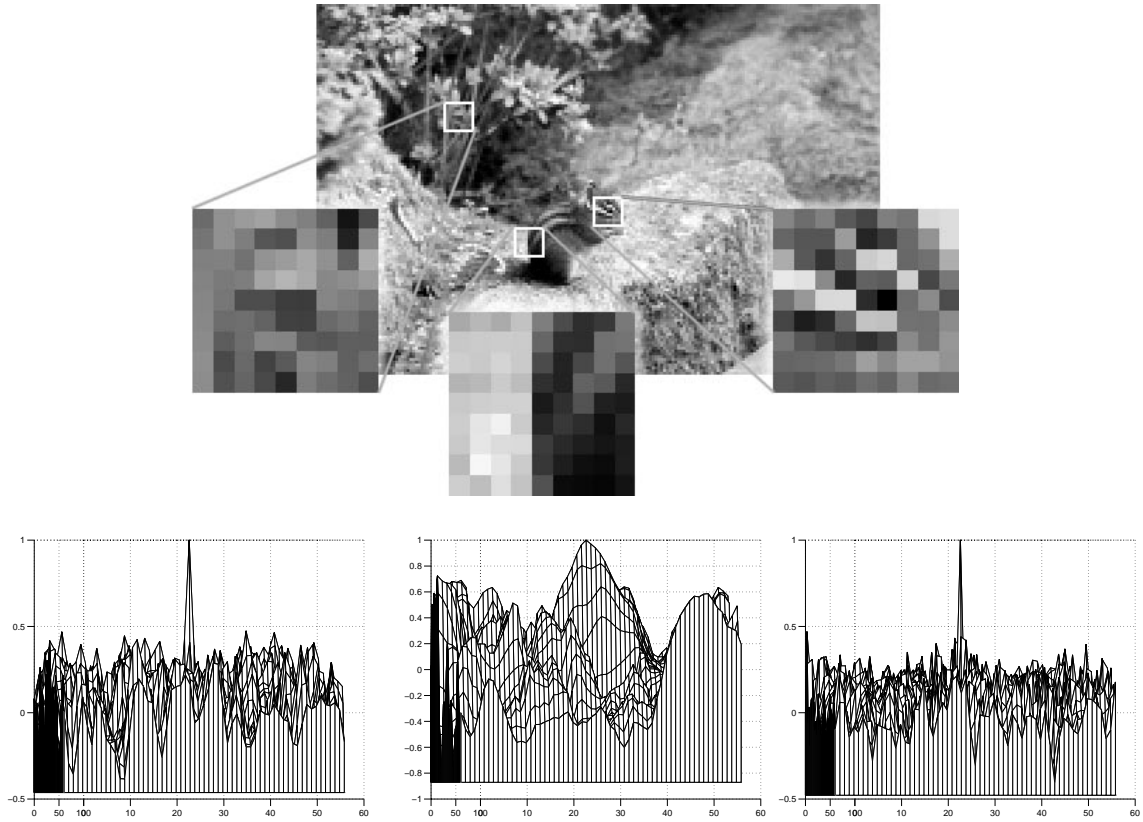[1] As stated earlier we use neighborhood sizes from 5×5 to 11×11.

Figure 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.
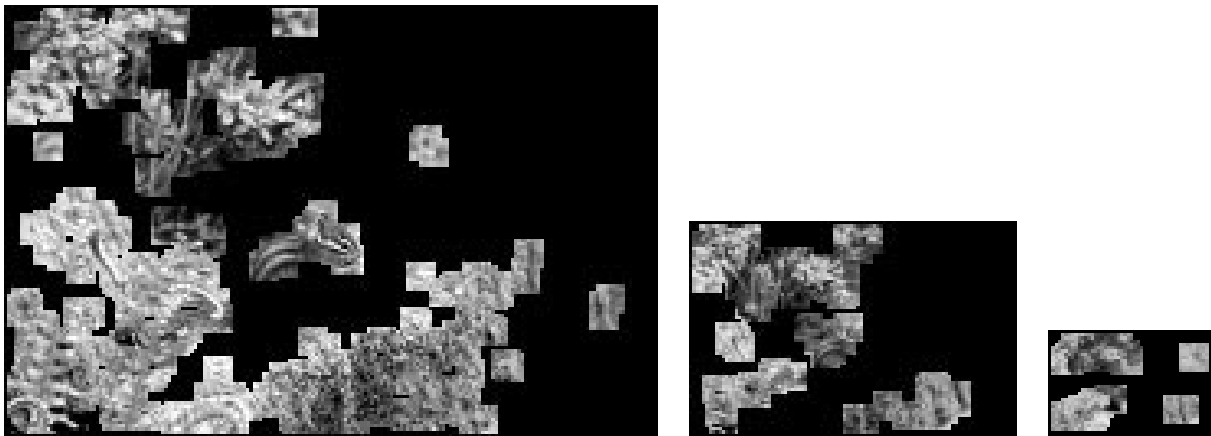

Figure 3: Selected feature points for 1/2, 1/4, and 1/8 resolutions; each point is shown with its 9×9 neighborhood.

olutions. It can be seen 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.

4.2 Recognizing Images

In this section we assume that we are given a collection of unknown images $S_{I'}$ and a known image I. We seek an image $I' \in S_{I'}$ such that I' and I are similar up to some affine transform. We assume that for image I we have selected sets of feature points at multiple resolutions. In this section we describe a recognition procedure that is used to match a single unknown image I' with I. Note that the procedure can be repeated for all images in $S_{I'}$.

Our recognition method proceeds as follows.

- Create a multiple resolution pyramid representation of image I' reducing the image scale of I' by c to obtain images $P_{I'} = \{I' \equiv I'_1, I'_c, I'_{c^2}, \ldots, I'_{c^k}\}$, where the subscripts correspond to the scale reduction. In our experiments we use $c = 2$. Other values of c can also be used. [Note that our representation of image I using feature points at multiple resolutions can be viewed as a pyramid P_I in which we consider only the feature points of I obtained at the appropriate resolution (see Figure 3). The factor used for scale reduction is 2. Also, it is possible to create additional image pyramids for various rotations and reflections.]

- Starting with feature points of I obtained at the lowest resolution, try to find matches with images in $P_{I'}$. We use the normalized cross-correlation described in Section 3.3 for point matching. There are three possible cases:

  1. No matches are found: we reject image I' due to lack of similarity with image I.
  2. Matches are found with multiple images in $P_{I'}$. In this case we use feature points obtained at higher resolution(s) for verification and refinement.
  3. A significant number of point matches with some image $I'_r \in P_{I'}$ is found: we accept image I' as a match for I and proceed to recover the transform parameters between I' and I (see Section 5). Note that we can obtain the approximate scale of I' from the image scales of the matching resolutions of I and I'_r.

Figure 4 shows two images derived from the image in Figure 1a. Figure 4a was created by applying an affine transform (cropping, scaling, and rotating) to the original. Figure 4b was created by cropping the image in Figure 4a. Following the steps described in this section we created image pyramids for both of these images with the lowest resolution being 1/8 of the size of the images. When compared to our collection of images at the lowest resolution (1/8 of the original images), a significant number of point matches were found for the image from Figure 1a.
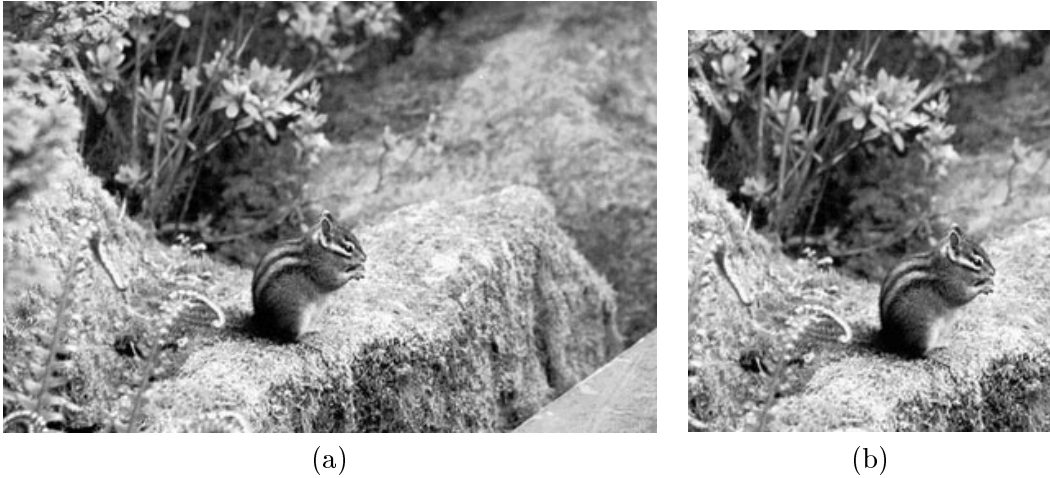
Figure 4: Recognizing images: (a) A distorted version of the image from Figure 1a. (b) A cropped version of image (a).

Additional confirmation was obtained by finding point matches at higher resolutions.

5   Recovering Watermarks

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

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

$$\left(\begin{array}{c} x'_i \\ y'_i \end{array} \right) = \left(\begin{array}{cc} a & b \\ c & d \end{array} \right) \left(\begin{array}{c} x_i \\ y_i \end{array} \right) + \left(\begin{array}{c} e \\ f \end{array} \right), \quad i = 1, \ldots, N. \tag{6}$$

We can rewrite equations (6) as

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

Equation (7) can be written as

$$\mathbf{A}\mathbf{u} = \mathbf{b} \tag{8}$$

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 [Ste73]

$$\mathbf{A}^T \mathbf{A} \mathbf{u} = \mathbf{A}^T \mathbf{b} = \mathbf{d}. \tag{9}$$

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

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

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:

$$\mathbf{A}_1^T \mathbf{A}_1 \mathbf{u}_1 = \mathbf{A}_1^T \mathbf{b}_1, \qquad \mathbf{A}_1^T \mathbf{A}_1 \mathbf{u}_2 = \mathbf{A}_1^T \mathbf{b}_2 \tag{11}$$

We solve these systems using the Cholesky decomposition [Ste73]. 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 frame $I^{(1)}$. [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'.]

<div align="center">(a)   (b)</div>

Figure 5: Recovering image size and aspect: (a) The recovered image from Figure 4a. (b) The recovered image from Figure 4b.

With regard to the reliability of the method two questions must be answered. The first question is geometrical and can be formulated as follows: Given the spatial distribution of the feature points in the image, which components of $\mathbf{u}$ can be computed? Note that if the feature points are distributed near a vertical line, x_is will be approximately constant and the first and third (as well as the fourth and sixth) columns of $\mathbf{A}$ will be linearly dependent. In such a case we would not be able to estimate all the components of $\mathbf{u}$. More generally, if the feature points are distributed along any line we would not be able to estimate $\mathbf{u}$ using the method described here. Fortunately, if such a situation occurs, the positive definite matrix $\mathbf{A}^T\mathbf{A}$ must have a large condition number (the ratio of its largest to its smallest eigenvalue); thus the existence of such situations is easy to detect by examining the eigenvalues of $\mathbf{A}^T\mathbf{A}$. Moreover, this observation can be used when deciding what feature points should be used for estimating the affine transform parameters between images I and I'.

The second question is numerical and can be formulated as follows: Given the spatial distribution and the orientations of the feature points in the image, and the accuracy with which the feature point correspondences can be computed (including rounding errors), how accurately can $\mathbf{u}$ be computed and what can be done to increase the numerical accuracy of the method? This question is also related to the condition number of $\mathbf{A}^T\mathbf{A}$ since the errors in the computed $\mathbf{u}$ are proportional to the errors in the feature point correspondences, where the condition number of $\mathbf{A}^T\mathbf{A}$ is the approximate proportionality coefficient. If the condition number is small we expect that the solution of (11) is reliable and that a few iterations of our algorithm will be enough to obtain a reliable estimate of u.

Figure 5 shows results of recovering the size and aspect of a distorted image using our method for the images in Figure 4. The estimated affine transform parameters $a - f$ for the image in Figure 4a

<div align="center">14</div>

were $(1.082 \quad 0.004 \quad -0.015 \quad 1.015 \quad -8.51 \quad 2.32)^T$; the corresponding inverse transform applied to the image resulted in the image shown in Figure 5a. Similarly, the estimated affine transform parameters for the image in Figure 4b were $(1.084 \quad 0.004 \quad -0.014 \quad 1.015 \quad -20.85 \quad -34.81)^T$; the corresponding inverse transform applied to the image resulted in the image shown in Figure 5b. Note that the recovered parameters are very similar for both the uncropped and cropped images. Note also that the embedded watermark [Dig] has been successfully detected in the recovered images.

5.2 Refinement Using Normal Displacement Fields

In the previous section we discussed various factors that may contribute to inaccuracies in estimating $\mathbf{u} = (a \quad b \quad e \quad c \quad d \quad f)^T$. However, it is possible to iteratively improve on the computed solution of the system (11) using the normal flow. 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 frame $I^{(1)}$. At this point we use normal flow to refine our estimate of the affine transform parameters $\mathbf{u}$.

The original image is used to estimate the normal displacement field (the normal flow) between I and $I^{(1)}$. The computed normal flow is then used to estimate the affine transform parameters $\mathbf{u}'$ between I and $I^{(1)}$. The estimated parameters are then used to correct $I^{(1)}$ and obtain the corrected frame $I^{(2)}$. If necessary, we then compute the normal flow between I and $I^{(2)}$ to make further refinements; however, typically we can stop after the first refinement step and use $I^{(2)}$ as our estimate of the watermarked image (before distortion). For a given estimate of $\mathbf{u}$ the stopping criterion is given by

$$\max_{(x,y) \in I} \{|\delta x|, |\delta y|\} < \varepsilon \tag{12}$$

where the maximum is computed using equation (2) over image I; typically we use $\varepsilon \leq 0.5$.

Typically, our estimates of $\mathbf{u}$ obtained using the method described in Section 4 are approximately correct. In most cases errors (see Equation (12)) are on the order of 1-2 pixels. To further refine our estimate of the original appearance of the distorted image I' we compute the normal flow between images I and $I^{(1)}$. Using Equation (2) we obtain the normal displacement field at (x, y) as

$$\delta \vec{r}_n \cdot \vec{n}_r = n_x \delta x + n_y \delta y = a_1 n_x x + b n_x y + e n_x + c n_y x + d_1 n_x y + f n_y \equiv \mathbf{a} \cdot \mathbf{u} \tag{13}$$

where $\vec{n}_r = n_x \vec{i} + n_y \vec{j}$ is the gradient direction at (x, y), $\mathbf{a} = (n_x x \quad n_x y \quad n_x \quad n_y x \quad n_y y \quad n_y)^T$, and $\mathbf{u}$ is the vector of affine parameters defined earlier. We use the method described in Section 3.4 to compute normal flow. For each edge point $\vec{r}_i$ we have one normal flow value $u_{n,i}$ which we use as the estimate of the normal displacement at the point. This gives us one approximate equation
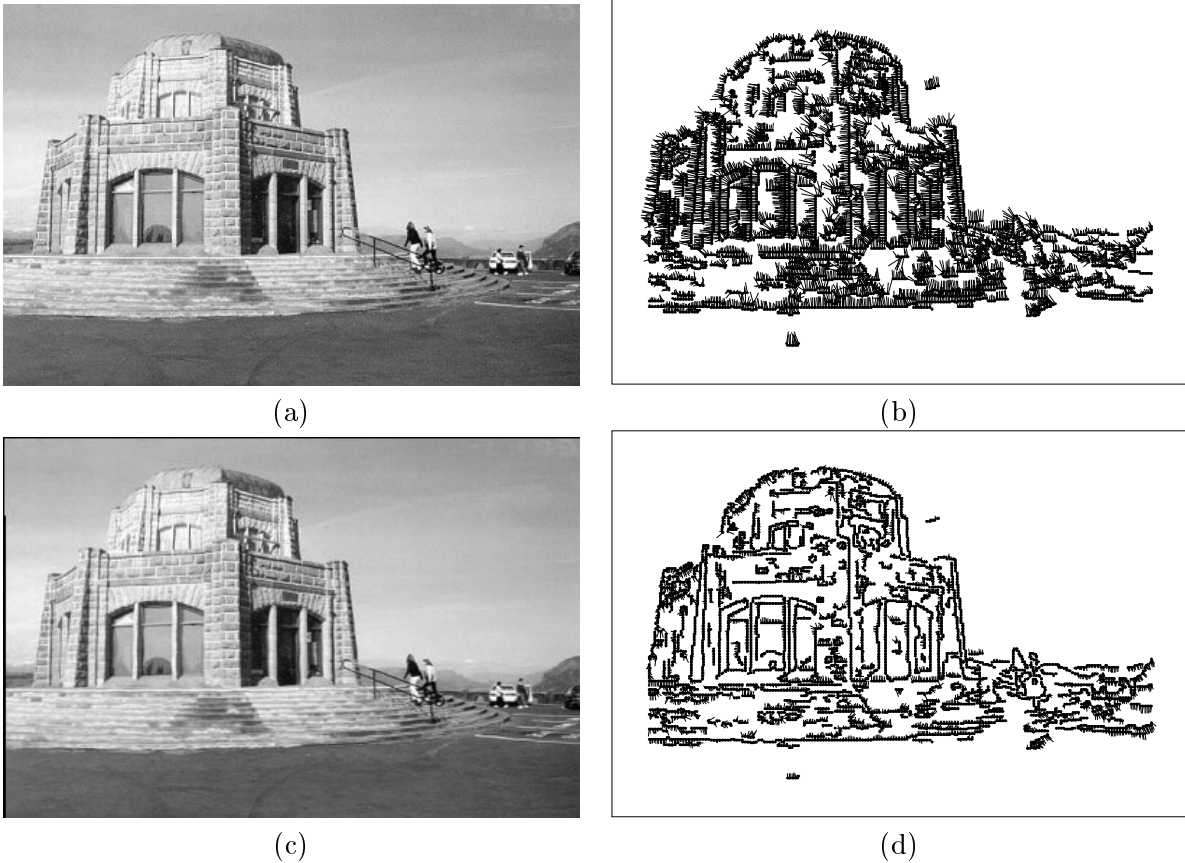
Figure 6: Refining image size and aspect: (a) The original image. (c) The normal displacement between the original and the distorted image (not shown). (c) The recovered image. (d) The normal displacement field between images (a) and (c).

$\mathbf{a}_i \cdot \mathbf{u} \approx u_{n,i}$. Let the number of edge points be $N \geq 6$. We then have a system

$$\mathbf{A}\mathbf{u} - \mathbf{b} = E$$

where $\mathbf{u}$ is an N-element array with elements $u_{n,i}$, $\mathbf{A}$ is an $N \times 6$ matrix with rows $\mathbf{a}_i$, and E is an N-element error vector. We seek $\mathbf{u}$ that minimizes $\|E\| = \|\mathbf{b} - \mathbf{A}\mathbf{u}\|$; the solution satisfies the system $\mathbf{A}^T\mathbf{A}\mathbf{u} = \mathbf{A}^T\mathbf{b}$ and corresponds to the linear least squares solution. As regards the reliability of the method we can make a similar argument as in our discussion of the reliability of the method described in Section 4. The positive definite matrix $\mathbf{A}^T\mathbf{A}$ must have a small condition number (the ratio of its largest to its smallest eigenvalue) for our method to work properly. Fortunately, this is easy to check and can be easily assured.

Figure 6 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

Figure 7: Additional images used in our experiments.

applying the recovery method described in the previous section.] Figure 6a shows the original image. Figure 6b 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 6c shows the recovered image that was obtained by applying the inverse affine transform to the distorted image. Finally, Figure 6d shows the normal displacement field between the images in Figure 6a and Figure 6c. 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.

5.3 Additional Experiments

We have experimented with the images shown in previous sections and with the images shown in Figure 7 using both a commercially available watermarking tool [Dig] and the watermarking technique described in [JDJ99]. A demo of the commercial watermark is available with Adobe Photoshop. The technique described in [JDJ99] 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 8.

Figure 8 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 8a). The watermark is not visible, but the enhanced image difference reveals it (see Figure 8b). An attack on the watermark is conducted by applying Stirmark against the watermarked image (see Figure 8c). Figure 8d 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
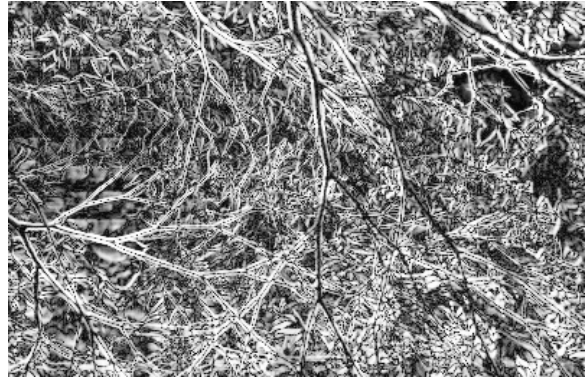
17

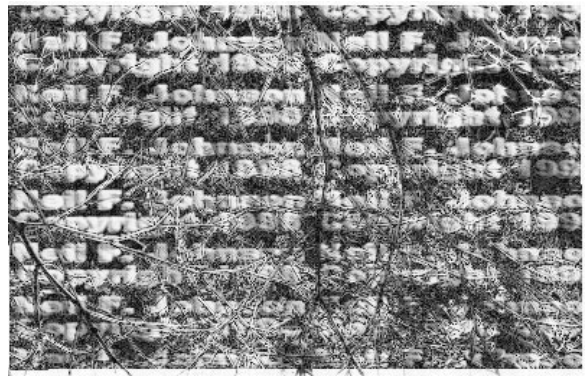Figure 8: Watermark recovery for the first image in Figure 7. (a) The watermarked image. (b) The enhanced difference between the original and the watermarked images. (c) The distorted image. (d) The enhanced difference between the original and the distorted images (no watermark is visible). (e) The recovered image. (f) The enhanced difference between the original and recovered images, revealing the watermark.

-0.0049 1.0045 0.9685 1.0939$)^T$. The recovered image is shown in Figure 8e. Finally, Figure 8f shows the enhanced difference between the recovered image and the original image, revealing the watermark.

6   Conclusion and Future Directions

Digital works are subject to illicit copying and distribution. The owners of such works are cautious in making them available without some means of identifying ownership and copyright. Digital watermarks provide mechanisms to embed and track the copyright and ownership of electronic works. Many techniques for watermarking of digital images have appeared in the recent literature; however, such embedded watermarks may fail due to accidental corruption or attack by cropping and/or affine distortions (e.g., rotation and scaling) [JJ98b, PAK98]. This hampers the ability to locate and identify watermarked images over distributed networks such as the Internet.

Understanding and investigating the limitations of watermarking applications can help direct researchers to better, more robust solutions to ensure the survivability of embedded information as well as to develop counter-measures such as alternatives to image recognition, recovery, and refinement. Methods that test the survivability of watermarks are essential for the development of stronger watermarking techniques [JJ98b, PAK98].

In this paper we provided a method for recognizing images, based on inherent features within images that can be used as identification marks. These identification marks can be applied to locate images and recover image size and aspect from distorted images. We provided examples showing that it is possible to recognize distorted images and recover their original appearances. In many cases doing so results in the recovery of embedded watermarks. The next phase of our research is to consider alternatives and improve the efficiency of our recognition and recovery techniques. Further work is required to enhance these proposed solutions and further investigate the modes and requirements for digital watermarking.

References

[And96] Anderson, R., (ed.), Information Hiding: First International Workshop, Cambridge, UK. Lecture Notes in Computer Science, Vol. 1174. Springer-Verlag, 1996.

[And96b] Anderson, R., Stretching the Limits of Steganography. In [And96], 39–48, 1996.

[AP98] Anderson, R., Petitcolas, F., "On the Limits of Steganography", *IEEE Journal on Selected Areas in Communications*, 16(4): 474–481, 1998.

[BGML96] Bender, W., Gruhl, D., Morimoto, N., Lu, A., "Techniques for Data Hiding", *IBM Systems Journal*, 35(3-4): 313–336, 1996.

[BLMO94] Brassil, J., Low, S., Maxemchuk, N., O'Gorman, L., Electronic Marking and Identification Techniques to Discourage Document Copying. In Infocom94, 1994. ftp://ftp.research.att.com/dist/brassil/1994/infwocom94a.ps.Z.

[BOML95] Brassil, J., O'Gorman, L., Maxemchuk, N., Low, S., Document Marking and Identification Using Both Line and Word Shifting. In Infocom95, Boston, MA, April 1995.

[Brau97] Braudaway, G.W., Protecting Publicly-Available Images with an Invisible Image Watermark. In [ICIP97], 1997.

[BS95] Brown, W., Shepherd, B.J., *Graphics File Formats: Reference and Guide*. Manning Publications, Greenwich, CT, 1995.

[Car95] Caronni, G., Assuring Ownership Rights for Digital Images, In *Reliable IT Systems*, Vieweg Publications, Wiesbaden, 1995.

[CKLS95] Cox, I., Kilian, J., Leighton, T., Shamoon, T., Secure Spread Spectrum Watermarking for Multimedia. Technical Report 95-10, NEC Research Institute, 1995.

[CKSL96] Cox, I., Kilian, J., Shamoon, T., Leighton, T., A Secure, Robust Watermark for Multimedia. In [And96], 185–206, 1996.

[CL97] Cox, I., Linnartz, J., Public Watermarks and their Resistance to Tampering. In [ICIP97], 1997.

[CMYY98] 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.

[Crav98] Craver, S., On Public-Key Steganography in the Presence of an Active Warden. In [IHW98], 1998.

[Dig] Digimarc Corporation, PictureMarc™, MarcSpider™, http://www.digimarc.com

[FB97] Ford, W., Baum, M., Secure electronic commerce. Upper Saddle River, NJ, Prentice Hall, 1997.

[FH97] Fleet, D., Heeger, D., Embedding Invisible Information in Color Images. In [ICIP97], 1997.

[GB98] Grhul, D., Bender, W., Information Hiding to Foil the Casual Counterfeiter. In [IHW98], 1998.

[GW92] Gonzalez, R.C., Woods, R.E., *Digital Image Processing*. Addison-Wesley. Reading, MA, 1992.

[HW96] Hsu, C, Wu, J., Hidden Signatures in Images. In [ICIP96], 1996.

[HYQ98] Hou, S., Yvo, D., Quisquater, J., Cerebral Cryptography. . In [IHW98], 1998.

[ICIP96] IEEE International Conference on Image Processing, Lausanne, Switzerland, 1996.

[ICIP97] IEEE International Conference on Image Processing, Santa Barbara, CA, 1997.

[IHW98] Second Information Hiding Workshop, Portland, Oregon, 1998.

[JDJ99] Johnson, N.F., Duric, Z., and Jajodia, S., "A Role for Digital Watermarking in Electronic Commerce", *ACM Computing Surveys*, 1999.

[JJ98] Johnson, N.F., Jajodia, S., "Exploring Steganography: Seeing the Unseen", *IEEE Computer*, 31(2): 26–34, 1998.

[JJ98b] Johnson, N.F., Jajodia, S., Steganalysis of Images Created using Current Steganography Software. In [IHW98], 1998.

[KH97] Kundur, D., Hatzinakos , D., A Robust Digital Image Watermarking Method Using Wavelet-Based Fusion. In [ICIP97], 1997.

[KRZ94] Koch, E., Rindfrey, J., Zhao, J., Copyright Protection for Multimedia Data. Proceedings of the International Conference on Digital Media and Electronic Publishing, Leeds, UK, 1994.

[Kuh97] M. Kuhn, F. Petitcolas, StirMark, http://www.cl.cam.ac.uk/~fapp2/watermarking/image_watermarking/stirmark, 1997.

[KZ95] Koch, E., Zhao, J., Towards Robust and Hidden Image Copyright Labelling. Proceedings of the 1995 IEEE Workshop on Nonlinear Signal and Image Processing, Neos Marmaras, Greece, 452–455, 1995.

[LMP98] Lach, J., Mangione-Smith, W., Potkonjak, M., Fingerprinting Digital Circuits on Programmable Hardware. In [IHW98], 1998.

[ODB96] O'Ruanaidh, J., Dowling, W., Boland, F., Phase Watermarking of Digital Images. In [ICIP96], 1996.

[OP97] O' Ruanaidh, J., Pun, T., Rotation, Scale and Translation Invariant Digital Image Watermarking. In [ICIP97], 1997.

[PAK98] Petitcolas, F., Anderson, R., Kuhn, M., Attacks on Copyright Marking Systems. In [IHW98], 1998.

[PBBC97] Piva, A., Barni, M., Bartolini, F., Cappellini, V., DCT-Based Watermark Recovering Without Resorting to the Uncorrupted Original Image. In [ICIP97], 1997.

[Rho97] Rhodes, G.B., "Steganography Methods Employing Embedded Calibration Data", United States Patent 5636292, 1997.

[SC96] Smith, J., Comiskey, B., Modulation and Information Hiding in Images. In [And96], 207–226, 1996.

[Sch96] Schneier, B., Applied Cryptography, Second Ed. John Wiley & Sons, New York, 1996.

[Sig] Signum Technologies, SureSign, http://www.signumtech.com/

[Ste73] G. W. Stewart. *Introduction to Matrix Computations*. Academic Press, New York, 1973.

[SZT96] Swanson, M., Zhu, B., Tewfik, A., Transparent Robust Image Watermarking. In [ICIP96], 1996.

[TNM90] Tanaka, K., Nakamura, Y., and Matsui, K., Embedding Secret Information into a Dithered Multi-level Image. Proceedings, IEEE Military Communications Conference, 216–220, 1990.

[Unz] Anonymous: unZign, http://altern.org/watermark/, 1997.

[VNP98] Voyatzis, G., N. Nikolaides, N., and Pitas, I., Digital Watermarking: An Overview, In *Proc. EUSIPCO'98*, Rhodes, Greece, September 1998.

[Wei93] Weiss, I., "Review: Geometric Invariants and Object Recognition". *International Journal of Computer Vision*, 10(3):207–231, 1993.

[XBA97] Xia, X., Boncelet, C.G., Arce, G.R., A Multiresolution Watermark for Digital Images. In [ICIP97], 1997.