Robust Image Watermarking Against Different Attacks: Noising, (Jpeg) Compression, And Filtering

Ahmed Hassan Hadi
Technical College of Najaf

Abstract
Many electronic watermarks for still images are sensitive to geometric distortion. For example, noising, JPEG compression and filtering of an image can prevent blind detection of a public watermark. In this paper, we propose a watermarking algorithm that the watermark is embedded into a (2-D) coefficients obtained by taking DCT of the image is robust to noising, JPEG compression and filtering distortions.

Keywords: Image watermarking, robustness, DCT, watermarking attacks

I. Introduction
The process of embedding a watermark in a multimedia object is termed as watermarking. Watermark can be considered as a kind of a signature that reveals the owner of the multimedia object. Content providers want to embed watermarks in their multimedia objects (digital content) for several reasons like copyright protection, content authentication, tamper detection etc.

(Maxemchuk et al., 1997), (Low et al., 1998), (Low et al., 1995) have published a series of papers on document watermarking based on line and word shifting. These methods are applicable to documents that contain paragraphs of printed text. Data is embedded in text documents by shifting lines and words spacing by a small amount (1/150 inch.) For instance, a text line can be moved up to encode a ‘1’ or down to encode a ‘0’, a word can be moved left to encode a ‘1’ or right to encode ‘0’. The techniques are robust to printing, photocopying, and scanning.

(Hartung et al., 1999), and (Seitz et al., 2005).
- Copyright protection embeds the copyright owner information in the image. This information is used in order to prevent others from alleging ownership of the image. This is the most common use of watermarking applications currently.
- In an image authentication application the intent is to detect alterations to the data. The image’s properties, such as it’s pixel averages, maximums and minims, are embedded and compared with the current images for modifications.
- Prevention of unauthorized copying is accomplished by embedding information about how often an image can be legally copied.
The fingerprint embeds information about the legal receiver in the image. This involves embedding a different watermark into each distributed image and allows the owner to locate and monitor pirated images that are illegally obtained. (Hwang, et al., 2000) proposed most significant bit watermarking embedding (MWE) scheme that uses visual cryptography for generation of shares based on the pixel value of the binary watermark and most significant bit of pixel value of the image (Hwang, et al., 2000), (Sleit, et al., 2008), and (Surekha, et al., 2011).

There are few algorithms that modify these DFT magnitude and phase coefficients to embed watermarks. (Ruaidh et al., 1996) proposed a DFT watermarking technique in which watermark is embedded by modifying the phase information within the DFT. It has been shown that phase based watermarking is robust against image contrast operation. Later (Ruaidh et al., 1998) showed how Fourier Mellin transform can be used for digital watermarking. Fourier Mellin transform is similar to apply Fourier transform to log-polar coordinate system for an image. This scheme is robust against against RST attacks.

(De Rosa et al., 1999) propose a scheme to insert watermark by directly modifying the mid frequency bands of the DFT magnitude component. (Ramkumar et al., 1999) also present a data hiding scheme based on DFT, where they modify the magnitude component of the DFT coefficients. Their simulations suggest that magnitude DFT survives practical compression, and this can be attributed to the fact that most practical compression schemes try to maximize the PSNR. Hence using magnitude DFT is a way to exploit the hole in most practical compression schemes. The proposed technique is shown to be resistant to JPEG and SPIHT compression.

(Lin et al., 2001) present a RST resilient watermarking algorithm. The watermark is embedded in the magnitude coefficients of the Fourier transform re-sampled by log-polar mapping. The technique is however not robust against cropping and shows weak robustness against JPEG compression (QF 70).

There is no a universal set of requirements that all watermarking technique must satisfy. Nevertheless, some general directions can be given for most of the applications (Podilchuk et al., 2001), (Cox et al., 2002), and (Katzenbeisser et al., 2000):

1. Perceptual transparency: in most applications, embedding a watermark should not affect the quality of the original media.
2. Robustness: watermarking can be robust, fragile and semi-fragile. Robust watermarking is employed to protect and verify the copyright; the purpose of fragile watermarking is content authentication for integrity and reality; and semi-fragile is used to distinguish between “information preserving” lossy transformations, such as compression, and “information altering” lossy transformations, such as replacement of objects (Podilchuk et al., 2001). Robustness is an important parameter for the watermarking detector with different strength. For example, a watermarking scheme is defined as robust if the detector can successfully detect the watermark in the processed image (Piper et al., 2009). Also, more precisely, robustness can be defined as the degree of resistance of a watermarking scheme to modifications of the host signal due to either common signal processing, or operations devised specifically in order to render the watermark undetectable (Tefas et al., 2009).
3. Capacity: the amount of watermark bits that can be stored in the host media.
4. Security: two levels of security can be identified (Katzenbeisser et al., 2000). At the higher level of security, an unauthorized party can neither detect the existence of
the embedded watermark, nor extract it. At the lower level, people can detect if the media are watermarked, however, the watermark cannot be extracted.

II. Insertion and Detection Algorithm
Figure (1) shows the insertion and detection watermarking algorithm.

Insertion Algorithm

1. Reading an image with size (512×512).
2. Take (256 × 256) discrete cosine transform of image size (512×512).
3. Take any two letters as the watermark
4. Convert the letters to gray level image with size (256×256).
5. multiply the watermark image with factor and the result plus one (x = β w + 1), let β = 10.
6. multiply the new watermark by dct's coefficients [by using block process in MATLAB]
8. Image watermarked applied to one of attacks "noise or compression (JPEG) or filtering"

Detection Algorithm

1. Take (256 × 256) discrete cosine transform of watermarked image.
2. Take (256 × 256) discrete cosine transform of original image.
3. divide coefficients of step1 on coefficients of step2 to get watermark as image (512×512).
Figure (1) The proposed watermarking algorithm

The proposed watermarking algorithm involves the following steps:

1. **Any watermark Letters as TV**
2. **Convert letters to gray level image "W"**
3. **New Watermark "β W+1"**
4. **Embedding**
5. **Original Image 512×512**
6. **Block Process DCT2**
7. **Watermarked Image**
8. **Attacks (Noise or compression or filtering)**
9. **Watermarked Image After attacks**
10. **Extracted Watermark Image**
III. Experimental results

A gray image of size 512×512 is analyzed using discrete cosine transform (DCT), the watermark is embedded in the DCT coefficients, and results are given in figures below.

The watermark that is embedding in images is shown in Figure (2). the experiments using the gray-level images and watermarked images shown in Figure (3). Figure (4), and (7) shows the effect of an attack based on worst case noising and its extracted watermark for each case.

The results for JPEG compression are displayed in Figure (5), and (8) for different quality factor (50%, 20%, 10%) and the watermark is detected even for a quality factor 10%.

Figure (6), and (9) show the results for unsharp, negative values, and wiener filtering, respectively. That applied on the watermarked images to attempt to destroy the watermark, and the proposed algorithm can extract the watermark for each case.

Robust against adding noise, robust against JPEG compression, and robust against filters illustrated as shown in Table (1)

<table>
<thead>
<tr>
<th>Attacks</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adding noise</td>
<td>0.2%, 2%, and 10%</td>
</tr>
<tr>
<td>Compression JPEG</td>
<td>50%, 20%, and 10%</td>
</tr>
<tr>
<td>Filtering</td>
<td>Unsharp, negative values, Wiener</td>
</tr>
</tbody>
</table>
Figure (2) Watermark

Figure (3) An original image a, c (left column) and its watermarked image b, d (right column).
Exp.1: Robust against adding noise

(a) Adding noise, 0.2% and its extracted watermark

(b) Adding noise, 2% and its extracted watermark

(c) Adding noise, 10% and its extracted watermark

Figure (4) Adding noise, (a) 0.2%, (b) 2%, (c) 10% and its extracted watermark for each case.
Robust against JPEG compression

(a) JPEG compression 50% and its extracted watermark

(b) JPEG compression 20% and its extracted watermark

(c) JPEG compression 10% and its extracted watermark

Figure (5) JPEG compression, (a)50%, (b)20%, (c)10%, and its extracted watermark for each case.
Robust against filters

(a) Unsharp filter and its extracted watermark.

(b) Negative values filter and its extracted watermark.

(c) Wiener filter and its extracted watermark.

Figure (6) Filters, (a) Unsharp, (b) Negative values, (c) Wiener, and its extracted watermark for each case.
Exp. 2: Robust against adding noise

(a) Adding noise, 0.2% and its extracted watermark

(b) Adding noise, 2% and its extracted watermark

(c) Adding noise, 10% and its extracted watermark

Figure (7) Adding noise, (a) 0.2%, (b) 2%, (c) 10% and its extracted watermark for each case.
Robust against JPEG compression

(a) JPEG compression 50% and its extracted watermark

(b) JPEG compression 20% and its extracted watermark

(c) JPEG compression 10% and its extracted watermark

Figure (8) JPEG compression, (a)50%, (b)20%, (c)10%, and its extracted watermark for each case.
Robust against filters

(a) Unsharp filter and its extracted watermark.

(b) Negative values filter and its extracted watermark.

(c) Wiener filter and its extracted watermark.

Figure (9) Filters, (a) Unsharp, (b) negative values, (c) Wiener, and its extracted watermark for each case.
Conclusion

This paper describes an algorithm for recovering watermark as image ("512 × 512" pixel). Any two letters convert to image ("256 × 256" pixel) as watermark is embedded in the DCT coefficients to get invisible watermarking for image protection. Experimental results demonstrate that this algorithm is robust against spatial attacks including noising, JPEG compression and filtering.

References


