Robust and Oblivious Image Watermarking Algorithm Based on Discrete Cosine Transform

B. Rakesh Reddy¹, Dr. K. Ramanjaneyulu², Dr. P. Pandarinath³, G. Nirmala⁴

¹Computer Science and Engineering Department, Sir CRR College of Engineering, India
²Electronics and Communication Engineering Department, PVP Siddhartha Institute of Technology, India
³Computer Science and Engineering Department, Gudlavalleru Engineering College, India
⁴Computer Science and Engineering Department, Sir CRR College of Engineering, India

Abstract—In this paper, Robust and Oblivious Image Watermarking scheme based on Discrete Cosine Transform (DCT) coefficient relation is presented. The cover image is transformed into DCT domain 8×8 block processing. The watermark bits are embedded into the cover image by adjusting the relationship among a group of middle frequency coefficients. The same relation between the coefficients is used to extract the watermark bits. Performance of watermarking algorithm in terms of Peak Signal to Noise Ratio, Normalized Cross-Correlation Coefficient, Bit-Error-Rate and Similarity Factor is presented against various image attacks. The results show that performance of the proposed method is good with majority of common image processing attacks.

Keywords—Digital Watermark, Discrete Cosine Transform, Robust Jpeg compression and Filter attacks.

Introduction

In the recent years, the field of image watermarking has attracted researchers because of its use to provide protection against illegal copying or tempering. Digital watermarking is designed to insert a secret signature, called a watermark, directly into a digital cover image to register the ownership [1]-[2]. Then, the watermark can be extracted when the ownership of the image needs to be identified. The watermark in the watermarked image can be either visible or invisible. For visible watermarking, the watermark should be perceptually visible [3] whereas for invisible watermarking, the watermark should be imperceptible [4].

In applications like copyright protection, robustness and transparency characteristics of the watermarked image play an important role. Watermarking scheme should be robust against general image processing attacks such as JPEG compression, cropping, noisy and image filtering [5]-[6]. For maintaining transparency, watermarking should not cause distortion to decrease the visual quality of the cover image.

Watermarking methods can be classified as spatial domain methods [7] and frequency domain methods. Spatial domain algorithms are simple and data embedding capacity is more and vulnerable. In this paper, Discrete Cosine Transform (DCT) is used to process the cover image in the frequency domain based on robust and oblivious watermarking [8]-[10]. The term oblivious indicates that there is no need of usage of cover image in the process of extraction. The term robust refers that there is high possibility of extracting watermark with less error even when the watermarked image is subjected to attacks. The cover image is firstly transformed into DCT domain by 8×8 block DCT transform. The watermark bits are embedded into the cover image by adjusting the relationship among a group of middle frequency coefficients. Performance of the method is good against many attacks in terms of PSNR and NCC.

Remainder of the paper is organized as follows. Section I describes Watermarking Requirements. In section II, Proposed Watermarking Method is explained. Section III provides Simulation Results and Conclusions in Section IV.

I. WATERMARKING REQUIREMENTS

The main requirements for an acceptable quality [11] [12] of watermarking are as follows.

a) Imperceptibility: The watermark should not be easily noticed by simple visual inspection.

b) Key uniqueness: Different keys should produce different, statistically independent watermarks.

c) Non invertibility: It should not be computationally feasible to find the watermark by possessing a watermarked image.

d) Image dependency: A single key produces a single watermark; however, this watermark should be adapted to the image content.

e) Reliable detection: The watermark should be efficiently detected for any value of false alarm probability up to a certain threshold.

f) Robustness: The watermark should be efficiently detected after most common signal processing operations.

II. PROPOSED WATERMARKING METHOD

There are three major issues in connection with the proposed watermarking method. They are (A)
Coefficient selection, (B) Watermark Embedding procedure and (C) Watermark Extraction procedure.

A. Coefficient Selection

Transform the cover image into 8x8 DCT blocks. Here, Fig: 1 represents a single 8x8 DCT block which is selected from the cover image. Select the coefficients of four which are neighbor to each other. From Fig: 1 the selected four coefficients are defined in two colors. Red color coefficient is the selected coefficient i.e. coeff1 (i, j) from other three coefficients are ref1 (i, j), ref2 (i, j) and ref3 (i, j) which are represented by dark blue color and they acts as the references for the selected coefficient coeff1 (i, j).

B. Watermark Embedding Method

In order to authenticate the image successfully, the embedded information should be robust against different attacks.

The embedding process uses the proposed scheme and it utilizes the relationship among 4 selected coefficients from different blocks to embed the watermark. This section will illustrate the embedding process in detail. The block diagram of the watermark embedding method Fig: 2.

Following are the steps of embedding process:

1. Split the Original Image into 8x8 blocks, and then apply DCT transformation on each block. So the DCT coefficients of the original image can be obtained.

2. Select four neighboring 8x8 blocks from the transform coefficients and arrange coefficients in four blocks. Let the selected coefficient blocks are identified as coeff1, ref1, ref2, and ref3, respectively. The watermark embedding position can be deemed as a secret key. For a given embedding frequency position, pick up the coefficient from coeff1, ref1, ref2 and ref3, then randomly select one coefficient from these four coefficients for watermark embedding and the others served as reference.

3. Here, coeff1 is selected from the 4 coefficients and remaining ref1, ref2, and ref3 are used as references.
4. The watermark bit \( w(k) \) to be embedded into position \( (i, j) \) in \( 8 \times 8 \) block of coeff1, the watermark embedding can be described in following manner:

\[
\text{Compute } P = \frac{(\text{ref1}(i, j) + \text{ref2}(i, j) + \text{ref3}(i, j))}{2.0}, \text{ for } 32\times32 \text{ size watermark}
\]

\[
\text{Compute } P = \frac{(\text{ref1}(i, j) + \text{ref2}(i, j) + \text{ref3}(i, j))}{3.0}, \text{ for } 64\times64 \text{ size watermark.}
\]

Where, \( P \) is used for modulation purpose for both sizes (32\times32 & 64\times64) of watermark.

By using this \( P \) we modulate the cover image as follows. Here (alpha=a)

\[
\text{coeff1}(i, j) = \begin{cases} 
P + a \text{ if } w(i, j) = 1 \text{ and coeff1}(i, j) < P, \\
P - a \text{ if } w(i, j) = 0 \text{ and coeff1}(i, j) > P
\end{cases}
\]

5. Coefficients post processing. Check every embedding position \((i,j)\), and process every embedding position as follows: Here (delta= d)

\[
\text{coeff1}(i, j) = \begin{cases} 
P - d \text{ if coeff1}(i, j) < P \text{ and } (P - \text{coeff1}(i, j)) < \delta, \\
P + d \text{ if coeff1}(i, j) > P \text{ and } (\text{coeff1}(i, j) - P) < \delta
\end{cases}
\]

Where, \( d \) is delta is another watermark embedding strength parameter.

Repeat step II, step IV, step V until all data bits are embedded and then perform inverse DCT transform on each processed block to obtain the watermarked image.

C. Watermark Extraction Method

The watermark extraction process does not require the original image, i.e. the proposed scheme is oblivious. The watermark embedding position and the block selection sequence during watermark embedding are utilized to extract the watermark.

The following Fig: 3 describe the extraction process:

Now we see extraction process in stepwise below:

1. Transform the watermarked image into DCT domain by 8\times8 block DCT.

2. Select four neighbor 8\times8 blocks from the transform coefficients, denoted as coeff1, ref1, ref2, and ref3. For a given embedding frequency position, pick up the coefficient selected that is coeff1 \((i, j)\). Randomly select one coefficient from these four coefficients for watermark extraction and the other served as reference. The watermark bit should be extracted from position \((i, j)\) in 8\times8 block of coeff1 \((i, j)\), the watermark extraction can be depicted as follows:

\[
\text{Use } P \text{ in extraction process which is calculated in embedded process i.e.} \\
\text{P = (ref1}(i, j) + ref2(i, j) + ref3(i, j))/2.0 \text{ for 32\times32 size watermark} \\
\text{P = (ref1}(i, j) + ref2(i, j) + ref3(i, j))/3.0 \text{ for 64\times64 size watermark}
\]

3. Repeat step II until all the watermark bits is extracted.

4. The decoded watermark bits are organized into m\times n matrix form \( W' \). Compare the extracted watermark with the original watermark \( W \). The bit error rate (BER) formula can be defined as follows:

\[
\text{BER}(W, W') = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} |W(i, j) - W'(i, j)|}{mn}
\]
Where, $\oplus$ is XOR operation. The extracted watermark can be compared by human eyes or decided by the value of BER.

5. The peak signal-to-noise ratio (PSNR) is used to evaluate the quality of the watermarked image in comparison with the cover image image. PSNR Formulas is as follows:

$$\text{PSNR} = 10 \log_{10} \frac{255 \times 255}{\frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} [f(i,j) - g(i,j)]^2} \text{ dB}$$

Where, m and n are the height and width of the image, respectively. $f(i, j)$ and $g(i, j)$ are the pixel values located at coordinates (i, j) of the original image, and the attacked image, respectively.

6. After extracting the watermark, the normalized correlation coefficient (NCC) is computed using the original watermark and the extracted watermark to judge the existence of the watermark and to measure the correctness of an extracted watermark. It is defined as

$$\text{NCC} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} [w(i,j) - w_{\text{mean}}][w'(i,j) - w'_{\text{mean}}]}{\sqrt{\left(\sum_{i=1}^{m} \sum_{j=1}^{n} [w(i,j) - w_{\text{mean}}]^2\right) \left(\sum_{i=1}^{m} \sum_{j=1}^{n} [w'(i,j) - w'_{\text{mean}}]^2\right)}}$$

Where, m and n are the height and width of the watermark, respectively. The symbols $(i,j)$ and $w'(i,j)$ are the bits located at the coordinates (i, j) of the original watermark and the extracted watermark respectively. The symbols $w_{\text{mean}}$ and $w'_{\text{mean}}$ are the values of the original watermark and the extracted watermark respectively.

7. The similarity factor is a metric which determines the similarity between the watermark inserted and the extracted watermark. The watermark can be extracted from simulated watermarked images with SM 1, 1 and 0.9991 respectively. SM is the similarity factor ranging from 0 to 1. SM=1 for the embedded watermark and the extracted watermark are same and any other value indicate the difference between them. Generally value of SM >0.75 is accepted for visible similarity. SM can be calculated using the equation follows.

$$\text{SM} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} w(i,j) \times w'(i,j)}{\sqrt{\left(\sum_{i=1}^{m} \sum_{j=1}^{n} w(i,j)^2\right) \left(\sum_{i=1}^{m} \sum_{j=1}^{n} w'(i,j)^2\right)}}$$

The expression has w and $w'$ is the original and the detected watermark and (i, j) is the coordinates of the coefficient that is selected.

2. SIMULATATED RESULTS

This paper has been executed on several standard 512×512 grayscale images, including Lena.bmp, Mountsea.bmp, scene.bmp and 32×32 pixels watermark image (both 32×32 pixels and 64×64 pixels) are used in this work. Both watermark images that are 32×32 pixels and 64×64 pixels are done in this paper, now we see the results for 32×32 pixels watermark which is embedded with cover image Lena.bmp (512×512 pixel).

![Fig: 4 Original image to Watermarked image](image)

Where, Fig: 4 describes that when Lena.bmp (cover image) watermarked with 32×32 pixels binary image (letters as image) in embedding process then we get the watermarked image. And the PSNR = 38.94 dB for cover image (lena.bmp where as watermarked image PSNR = 38.89 dB, where alpha=40 and beta=8.

When the watermarked image undergone some common image processing attacks, some simulated results are shown in Table 1 (Lena.bmp watermarked with 32×32(WI) image): shows different attacks which are undergone at quality factor. Other attacks on when the watermarked image undergone common image processing attacks, some simulated results are shown in Table 2. When the watermarked image is auto levels and auto contrast adjusted by Photoshop, the watermark can be extracted with bit errors.

When the watermarked image is rotated 1 degree and then rotated back, the watermark can be detected with bit errors 0.032. The proposed scheme has good localization ability when the watermarked image is cropped or tampered.

When the watermarked image is tampered by Gaussian noisy with density 0.0005, the watermark can be extracted with BER 0.04.To evaluate the robustness of the proposed approach, three popular testing images one of them is: Lena cover image were watermarked with the proposed method. Gaussians filtering and noisy were applied on the watermarked image. The results are shown in Table 2.

<table>
<thead>
<tr>
<th>QF of JPEG Compression</th>
<th>Extracted Watermark</th>
<th>PSNR of the attacked watermark</th>
<th>Quality of the Extracted watermark</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="image">Table 1 Simulated Results for JPEG Compression Attack on Watermarked image</a></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
When the watermarked image is Gaussian filtered with 5×5 and 7×7 filters, the results are shown in Table 2, respectively.

Table 2 Simulated Results for Other Attacks on Watermarked Image

<table>
<thead>
<tr>
<th>S.N</th>
<th>Type of the Attack</th>
<th>PSNR of the attacked watermarked image (dB)</th>
<th>BER</th>
<th>SM</th>
<th>NCC</th>
<th>Rotation</th>
<th>BER</th>
<th>SM</th>
<th>NCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Auto Level</td>
<td>21.36</td>
<td>CHN</td>
<td>0.0029</td>
<td>0.9981</td>
<td>0.9919</td>
<td>0.1936</td>
<td>0.8868</td>
<td>0.2203</td>
</tr>
<tr>
<td>2</td>
<td>Auto Contrast</td>
<td>21.36</td>
<td>CHN</td>
<td>0.0029</td>
<td>0.9981</td>
<td>0.9919</td>
<td>0.1936</td>
<td>0.8868</td>
<td>0.2203</td>
</tr>
</tbody>
</table>

Now we see Lena.bmp embedded with watermarked with 64×64 pixels binary image, where alpha=15 and delta=8 is used. And the PSNR = 35.26 dB and watermarked image PSNR = 35.23 dB. Below Fig: 5 show the both images.

![Fig: 5 Embedded Process from cover image to watermarked image](image)

The simulated results for quality factor this cover image listed in the Table 3. Quality Factor starts at 10 to 100 and these changes the PSNR, BER, SM and NCC.

Table 3 Simulated Results of JPEG Compression attack on Watermarked image

<table>
<thead>
<tr>
<th>QF of JPEG Compression Attack</th>
<th>Extracted Watermark image</th>
<th>PSNR(db) of attacked watermark image</th>
<th>Quality of Extracted watermark</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| 10                           |                           | 29.57                               | 0.1936 | 0.8868 | 0.2203 |
Other attacks on when the watermarked image undergone common image processing attacks, some simulated results are shown in Table 4. These simulated results show that the proposed scheme have good robustness.

Table 4 Simulated Results of other attacks on Watermarked image

<table>
<thead>
<tr>
<th>Types of Attack</th>
<th>PSNR(dB) of attacked watermark image</th>
<th>Extracted Watermark image</th>
<th>Quality of Extracted watermark</th>
<th>BER</th>
<th>SM</th>
<th>NCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto level</td>
<td>33.54</td>
<td>CHN</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Auto Contrast</td>
<td>33.54</td>
<td>CHN</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Rotation 1°</td>
<td>25.86</td>
<td>CHN</td>
<td>0.0388</td>
<td>0.9772</td>
<td>0.8558</td>
<td></td>
</tr>
<tr>
<td>Cropping</td>
<td>11.39</td>
<td>CHN</td>
<td>0.0310</td>
<td>0.9825</td>
<td>0.8659</td>
<td></td>
</tr>
<tr>
<td>Gaussian filtering(3,3) at density 0.2</td>
<td>35.26</td>
<td>CHN</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Gaussian filtering(5,5) at density 0.2</td>
<td>35.26</td>
<td>CHN</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Gaussian filtering(7,7) at density 0.2</td>
<td>35.26</td>
<td>CHN</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Gaussian Noise</td>
<td>30.81</td>
<td>CHN</td>
<td>0.0271</td>
<td>0.9841</td>
<td>0.8970</td>
<td></td>
</tr>
</tbody>
</table>

When the watermarked image is auto levels and auto contrast adjusted by Photoshop, the watermark can be extracted without bit errors. When the watermarked image is rotated 1 degree and then rotated back, the watermark can be detected with bit errors 0.038. The proposed scheme has good localization ability when the watermarked image is cropped or tampered.

When the watermarked image is tampered by Gaussian noisy with density 0.0005, the watermark can be extracted with BER 0.027. To evaluate the robustness of the proposed approach, three popular testing images one of them is: Lena cover image were watermarked with the proposed method.
Gaussians filtering and noisy were applied on the watermarked image. The results are shown in Table 4. When the watermarked images are Gaussian filtered with 3x3 filter, since the watermark can be detected without bit errors, the results are shown here. When the watermarked image is Gaussian filtered with 5x5 and 7x7 filters, the results are shown in Table 4, respectively. These Simulated results show that the proposed scheme have good robustness.

3. CONCLUSIONS

In this work, the watermark image of 64x64 size was watermarked and it is an extension of the existing method [4] and got better simulated results. The performance of proposed method for watermark image of 32x32 size was improved slightly in terms of PSNR and BER. Visual quality of both watermarked image and extracted watermark is good. Performance of the proposed method against various attacks is presented and is satisfactory.

REFERENCES