Hybrid DCT-CT Digital Image Adaptive Watermarking

Bijan Fadaeenia, Nasim Zarei
Electrical engineering department
Islamic Azad University-Hamedan Branch
Hamedan, Iran
Fadaeenia@iauh.ac.ir, Nasim.zarei@iauh.ac.ir

Abstract—This paper proposes a robust and blind digital image watermarking which uses a correlation based algorithm to embed a binary pseudo-random sequence into a grayscale host image. This scheme applies a combination of Discrete Cosine Transform (DCT) and Contourlet Transform (CT). Due to increasing the imperceptibility of embedded watermark, the power of watermark is varying in different regions of host image. The varying watermark power calculation is based on fuzzy decision. Experimental results show that the proposed method is robust against both geometric and non-geometric attacks.

Keywords—Watermarking; DCT; Contourlet; Fuzzy.

I. INTRODUCTION

Digital watermarking has been identified as a possible solution for Copyright protection of digital media and has become an area of increased research activity over the last decade [1]. Commonly, a digital watermark is a code that is embedded into a media. It plays the role of a digital signature, providing the media with a sense of ownership or authenticity. The primary benefit of watermarking is that the content is not separable from the watermark [2]. In the case of digital image this technique tries to embed invisible information in digital image. As mentioned before, the digital watermark must be robust against media manipulations [3].

Although most of works in the field of watermarking focuses on using the multiresolution analysis proposed by Wavelet Transform [4], [5], [6], [7], Contourlet Transform began to gain some interests for its capability of capturing directional information such as smooth contours, and directional edges [1], [8], [9], [10], [11], [12].

In this paper, we compute two level CT of host image then we divide selected sub-bands into blocks and apply DCT to each block. Next, we generate two pseudo-random uncorrelated sequences for embedding 0 and 1 and alter the CT-DCT coefficients by using a fuzzy system. We will show that the proposed algorithm can resist against both geometric and non-geometric attacks and increase the PSNR.

This paper is organized as follows: in Section 2, a quick view of CT and its advantages for watermarking will be provided. The proposed algorithm will be introduced in Section 3. Experimental results will be shown in Section 4. Finally, Section 5 concludes the paper.
Therefore, the change of HF coefficients does not affect the LF coefficients. Because the WT does not have the spreading effect as the LP, the embedded watermark is susceptible to the attacks such as low-pass filtering, quantization and compression that destroy the HF coefficients of the image seriously. In contrast, if the watermark is embedded into the largest detail subbands of CT, it is likely to be spread out into all subbands when we reconstruct the watermarked image. Thus, the watermarking scheme in CT domain may be robust to the widely spectral attacks resulting from both LF image processing and HF image processing.

III. PROPOSED ALGORITHM

In this paper, a 1225 bits pseudo-random sequence is used as watermark. In this algorithm sub-bands 2 and 3 in second level are selected (Fig. 2), because human eyes are less sensitive to noise in oblique orientation [18] and these subbands are the most oblique [13].

A. Watermark embeding

Step 1: Two level contourlet is applied to image and image is divided into nine sub-bands.

Step 2: Sub-bands 2 and 3 of second level are selected for watermark insertion.

Step 3: Selected sub-bands are divided into \( N \times N \) blocks and the DCT of each block is computed. These blocks are denoted as \( \text{block}_i \), \( i = 1, 2, ... M \) and \( M \leq N_w \), which \( N_w \) is the number of watermark bits. If dimensions of watermark logo are denoted as \( L_w \) and \( H_w \) then \( N_w = L_w \times H_w \). Table I, is shown that the combination of CT and DCT can increase the quality of watermarked image. Table II, is shown that the combination of CT and DCT can increase the robustness of algorithm against both geometric and non-geometric attacks by comparing NC values.

Step 4: If each element of watermark is denoted as \( W_j, j = 1, 2, ..., N_w \) Two uncorrelated pseudo-random sequences with zero mean are generated, one of them is used for embedding \( W_j = 0 \) and the other one for \( W_j = 1 \), which denoted as \( PN_0 \) and \( PN_1 \) respectively.

Step 5: To embed watermark into the produced blocks, the middle frequency coefficients of each block are selected and denoted as \( Y \). This selection is a tradeoff between robustness and imperceptibility of watermark.

Step 6: The optimum watermarking weight, \( \alpha \), is calculated for each block using a fuzzy system. The used fuzzy system will be explained later.

Step 7: The watermark bits are embedded into image as follows:

\[
Y' = Y + \alpha \cdot PN_0 - \alpha \cdot PN_1 \quad \text{if} \quad W = 0
\]

\[
Y' = Y + \alpha \cdot PN_1 - \alpha \cdot PN_0 \quad \text{if} \quad W = 1
\]

Step 8: By Applying inverse DCT (IDCT) to each block after its mid-band coefficients have been modified, the CT of watermarked image is generated.

Step 9: Finally the watermarked image can be produced by using inverse CT (ICT).

The embedding process is depicted in Fig. 3.

B. Watermark extraction

step 1: Two level contourlet is applied to watermarked image and watermarked image is divided into nine sub-bands.

step 2: Selected sub-bands are divided into \( N \times N \) blocks.

step 3: Two pseudo-random sequences \( PN_0 \) and \( PN_1 \) are regenerated by using same seeds as embedding stage.

step 4: For each block in the selected sub-band, correlations between mid-band coefficients with \( PN_0 \) and \( PN_1 \) are calculated. If the correlation with the \( PN_0 \) is higher than the correlation with \( PN_1 \), the watermark bit will be considered as \( 0 \), otherwise it will be considered as \( 1 \).

step 5: After watermark extraction, similarity between the original and extracted watermarks is computed.

The extraction process is depicted in Fig. 4.

C. Fuzzy system

The used fuzzy system is a two-input and one-output system. The system tries to balance watermark power and brings us both robustness and invisibility. This fuzzy system...
is based on a simple rule: In watermarking, large coefficients can be changed more than small ones. Two inputs of fuzzy system are named as “AVERAGE” and “DISTANCE”. AVERAGE is the average of each block $Y$ (watermark embedding: step 5).

$$AVERAGE_i = \text{average}(Y_i), i = 1...N_W$$  \hspace{1cm} (3)

$N_W$ is the number of watermark bits. To calculate the DISTANCE two correlation must be calculated. Correlation between $PN_0$ and $Y_i$ which named $\text{DIFF}_0$ and correlation between $PN_1$ and $Y_i$ which named $\text{DIFF}_1$. Calculation of DISTANCE is related to value of watermark bits:

$$DISTANCE_i = \text{DIFF}_0 - \text{DIFF}_1 \quad \text{if} \quad W = 0 \hspace{1cm} (4)$$

$$DISTANCE_i = \text{DIFF}_1 - \text{DIFF}_0 \quad \text{if} \quad W = 1 \hspace{1cm} (5)$$

$DISTANCE \in [-1.8, +1.8]$. This algorithm tries to increase the correlation of each $Y_i$ with $PN_0$ or $PN_1$ in respect of value of $W$, hence fuzzy rules of system can be written as below:

1: If AVERAGE is very small then $\alpha$ is Very small.
2: If AVERAGE is small then $\alpha$ is small.
3: If AVERAGE is medium then $\alpha$ is medium.
4: If AVERAGE is large then $\alpha$ is large.
5: If AVERAGE is very large then $\alpha$ is very large.
6: If DISTANCE is very small then $\alpha$ is very small.
7: If DISTANCE is small then $\alpha$ is small.
8: If DISTANCE is medium then $\alpha$ is medium.
9: If DISTANCE is large then $\alpha$ is large.
10: If DISTANCE is very large then $\alpha$ is very large.

The surface of fuzzy system is shown in Fig. 5.

### D. An example

Explanation of embedding and extracting of a watermark bit into the host image is the goal of this example. As shown in Fig. 6 a grayscale, $512 \times 512$ of cat image is selected as host image. In this example, the procedure of inserting a “0” as a watermark bit into the host image will be presented.

1) **Embedding:** First Two level contourlet is applied to image and image is divided into nine sub-bands as shown in Fig. 7.

Then subbands 2 and 3 of second level are selected for watermark insertion (Fig. 8). Next, the selected subbands are divided into $512 \times 512$ blocks. Hence, 625 blocks per subband is obtained. In this example, the $33^{th}$ bit of host image is selected for embedding.

DCT of selected block will be computed and its middle frequency will be selected. For further use it is named as
Mid.

\[
Mid = \begin{bmatrix}
2.9323 & 0.7020 \\
1.1322 & 1.1205
\end{bmatrix}
\] (6)

After that, watermark power is calculated with fuzzy system:

\[
Watermark\ power = 59.2577
\] (7)

Then \(PN_0\) and \(PN_1\) will be produced. By using equation (1), the new \(Mid (Mid^*)\) will be obtained:

\[
Mid^* = \begin{bmatrix}
-13.3630 & 16.9973 \\
-21.0886 & 23.3414
\end{bmatrix}
\] (8)

Finally, \(Mid\) will be replaced with \(Mid^*\) and all above steps will be done reversely to produce watermarked image.

Figure 9: Watermarked image of Cat

2) Extraction: Such as embedding section, middle frequency of 33\textsuperscript{th} block (\(WMid\)) is produced:

\[
WMid = \begin{bmatrix}
-4.4099 & 14.6900 \\
-20.0518 & 14.5569
\end{bmatrix}
\] (9)

By calculating the correlation between this matrix and \(PN_0\) and \(PN_1\), we will have:

\[
\text{Correlation (}(PN_0, WMid)\text{) = 0.1272}
\] (10)

\[
\text{Correlation (}(PN_1, WMid)\text{) = 0.0589}
\] (11)

By comparison of above results, the watermark bit will be acquired as "0".

IV. EXPERIMENTAL RESULTS

Testing this algorithm is done by using 512 \times 512 gray scale host images: Man, Pepper, Baboon, Lena, Gold hill, Plane, Boat and Cat. Only partial results of them are shown. Watermark \(W\) is a binary pseudo-random sequence with 1225 bits length. For brevity only the pepper image is shown here. Fig. 10.

To evaluate this algorithm, we used three factors: PSNR, MSSIM and normalized Correlation Coefficient (NC). PSNR is used to evaluate the quality of watermarked image regardless of HVS, but MSSIM tries to evaluate the image quality in respect to HVS. The used PSNR formula is [10]:

\[
PSNR = 10\log_{10} \frac{255 \times 255}{\sum_{x=0}^{L_w-1} \sum_{y=0}^{H_w-1} [f(x,y) - g(x,y)]^2}
\] (12)

where \(H_w\) and \(L_w\) are the height and width of the image, respectively. \(f(x,y)\) and \(g(x,y)\) are the values located at coordinates \((x,y)\) of the original image, and the watermarked image, respectively.

For the sake of brevity, explanation of MSSIM is omitted. It is available in [19].

After watermark extraction, the normalized Correlation Coefficient (NC) is computed using the original watermark and the extracted watermark to measure the correctness of an extracted watermark. It is defined as [10]:

\[
NC = \frac{\sum_{i=1}^{N_w} w_i \bar{w}_i'}{\sqrt{\sum_{i=1}^{N_w} w_i^2} \sqrt{\sum_{i=1}^{N_w} \bar{w}_i'^2}}
\] (13)

where \(N_w\) is the number of watermark bits. \(w\) and \(\bar{w}\) are the original watermark and the extracted watermark, respec-

Table I: PSNR and MSSIM of watermarked images. Combination of DCT and CT can increase the quality of watermarked image.

<table>
<thead>
<tr>
<th>Transform</th>
<th>Factor</th>
<th>Baboon</th>
<th>Cat</th>
<th>Man</th>
<th>Pepper</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT &amp; DCT</td>
<td>PSNR</td>
<td>40.82</td>
<td>41.83</td>
<td>41.3</td>
<td>42.13</td>
</tr>
<tr>
<td>CT &amp; DCT</td>
<td>MSSIM</td>
<td>0.990</td>
<td>0.985</td>
<td>0.984</td>
<td>0.988</td>
</tr>
<tr>
<td>CT</td>
<td>PSNR</td>
<td>38.48</td>
<td>40.13</td>
<td>40.45</td>
<td>39.85</td>
</tr>
<tr>
<td>CT</td>
<td>MSSIM</td>
<td>0.983</td>
<td>0.974</td>
<td>0.979</td>
<td>0.957</td>
</tr>
</tbody>
</table>

Table II: Comparison of NC values for CT and CD-DCT.

<table>
<thead>
<tr>
<th>Test</th>
<th>CT</th>
<th>CT-DCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian filter 5\times5</td>
<td>0.8858</td>
<td>0.9611</td>
</tr>
<tr>
<td>Median filter 5\times5</td>
<td>0.7417</td>
<td>0.8892</td>
</tr>
<tr>
<td>Average filter 5\times5</td>
<td>0.6503</td>
<td>0.9354</td>
</tr>
<tr>
<td>Gaussian noise (Var=20)</td>
<td>0.7367</td>
<td>0.7727</td>
</tr>
<tr>
<td>Salt and pepper (density=0.05)</td>
<td>0.8364</td>
<td>0.8353</td>
</tr>
<tr>
<td>JPEG (10%)</td>
<td>0.7125</td>
<td>0.7118</td>
</tr>
<tr>
<td>Scaling (50%)</td>
<td>0.9680</td>
<td>0.9848</td>
</tr>
<tr>
<td>Cropping (75%)</td>
<td>0.7066</td>
<td>0.7222</td>
</tr>
<tr>
<td>Histogram equalization</td>
<td>0.9768</td>
<td>0.9890</td>
</tr>
<tr>
<td>Image sharpening</td>
<td>0.9868</td>
<td>0.9912</td>
</tr>
</tbody>
</table>
The watermarked images and extracted watermark have been shown in Fig. 11.

In experiments both geometric and non-geometric attacks are considered. Non-geometric attacks includes JPEG compression, histogram equalization, sharpening and gaussian, median and average filtering and Gaussian noise, salt and pepper noise. For geometric attacks, scaling and cropping are used. The results are shown in Tables III, V, and IV. A comparison between proposed method and some previous works is shown in Table VI, in this comparison “Lena” image is used as host [10], [11], [12].

A group of 10,000 different watermarks including the genuine (embedded) one is used for evaluating the robustness of watermark detection algorithm against various attacks. As mentioned before each watermark is a 1225-bit binary pseudo-random sequence. The experimental results show that for genuine watermark the detector algorithm has highest response (the genuine watermark is 5000th watermark).

Detection responses of algorithm against various attacks and relevant images are shown in Fig. 11 and Fig. 12, respectively, and prove that proposed method is robust against both geometric and non-geometric attacks.

V. CONCLUSION

A blind watermarking scheme was proposed in this paper, which uses Contourlet and Discrete Cosine Transform to increase robustness and invisibility. By considering the fact that human eyes are less sensitive to noise in oblique angels, Contourlet was used because of its directional property. On

---

### Table III: Correlation Coefficients after attack by median, gaussian, Average filtering with various filter size \((n \times n)\), and histogram equalization

<table>
<thead>
<tr>
<th>Image</th>
<th>Median 3x3</th>
<th>Median 5x5</th>
<th>Median 13x13</th>
<th>Gaussian 3x3</th>
<th>Gaussian 5x5</th>
<th>Gaussian 13x13</th>
<th>Average 3x3</th>
<th>Average 5x5</th>
<th>Average 13x13</th>
<th>Hist equal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>0.9633</td>
<td>0.6703</td>
<td>0.9655</td>
<td>0.7930</td>
<td>0.9650</td>
<td>0.7280</td>
<td>0.9729</td>
<td>0.9729</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Man</td>
<td>0.9270</td>
<td>0.6799</td>
<td>0.9404</td>
<td>0.7630</td>
<td>0.9386</td>
<td>0.7062</td>
<td>0.9599</td>
<td>0.9599</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pepper</td>
<td>0.9834</td>
<td>0.6888</td>
<td>0.9880</td>
<td>0.8098</td>
<td>0.9879</td>
<td>0.7398</td>
<td>0.9890</td>
<td>0.9890</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baboon</td>
<td>0.9135</td>
<td>0.6789</td>
<td>0.9289</td>
<td>0.7765</td>
<td>0.9289</td>
<td>0.7062</td>
<td>0.9278</td>
<td>0.9278</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table IV: Correlation Coefficients after attack by JPEG compression with various quality and sharpening

<table>
<thead>
<tr>
<th>Image</th>
<th>JPEG Quality 10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>65</th>
<th>75</th>
<th>80</th>
<th>Sharpening</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>0.7797</td>
<td>0.8317</td>
<td>0.8654</td>
<td>0.8940</td>
<td>0.9243</td>
<td>0.9576</td>
<td>0.9689</td>
<td>0.9756</td>
<td>0.9790</td>
<td>0.9810</td>
<td>0.9857</td>
</tr>
<tr>
<td>Man</td>
<td>0.7591</td>
<td>0.8029</td>
<td>0.8452</td>
<td>0.8884</td>
<td>0.9066</td>
<td>0.9248</td>
<td>0.9311</td>
<td>0.9386</td>
<td>0.9445</td>
<td>0.9453</td>
<td>0.9513</td>
</tr>
<tr>
<td>Pepper</td>
<td>0.7118</td>
<td>0.7535</td>
<td>0.8310</td>
<td>0.8873</td>
<td>0.9165</td>
<td>0.9723</td>
<td>0.9846</td>
<td>0.9885</td>
<td>0.9881</td>
<td>0.9910</td>
<td>0.9912</td>
</tr>
<tr>
<td>Baboon</td>
<td>0.7928</td>
<td>0.8469</td>
<td>0.8943</td>
<td>0.9112</td>
<td>0.9135</td>
<td>0.9279</td>
<td>0.9348</td>
<td>0.9406</td>
<td>0.9500</td>
<td>0.9505</td>
<td>0.9366</td>
</tr>
</tbody>
</table>

### Table V: Correlation Coefficients after attack by gaussian noise which its variance varies from 1 to 10, salt & pepper noise which varies from 2% to 20%, cropping up to 75% of image and scaling 50% and 75%

<table>
<thead>
<tr>
<th>Image</th>
<th>Salt and Pepper</th>
<th>Gaussian noise</th>
<th>Cropping (%)</th>
<th>Scaling (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>0.9042 0.8424 0.7314</td>
<td>0.9366 0.7757 0.7218</td>
<td>0.9383 0.8346 0.7140</td>
<td>0.9582 0.9769</td>
</tr>
<tr>
<td>Man</td>
<td>0.8837 0.8244 0.7337</td>
<td>0.9031 0.7730 0.7277</td>
<td>0.9114 0.8248 0.6998</td>
<td>0.9294 0.9450</td>
</tr>
<tr>
<td>Pepper</td>
<td>0.9090 0.8353 0.7105</td>
<td>0.9320 0.7727 0.7192</td>
<td>0.9500 0.8452 0.7222</td>
<td>0.9848 0.9890</td>
</tr>
<tr>
<td>Baboon</td>
<td>0.8894 0.8218 0.7430</td>
<td>0.9044 0.7956 0.7334</td>
<td>0.9009 0.8184 0.7017</td>
<td>0.9223 0.9406</td>
</tr>
</tbody>
</table>

### Table VI: Comparison between proposed method and some previous works.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian filter 3x3</td>
<td>0.9321</td>
<td>0.7658</td>
<td>0.8956</td>
<td>0.9861</td>
</tr>
<tr>
<td>Median filter 3x3</td>
<td>NA</td>
<td>NA</td>
<td>0.9689</td>
<td>0.9868</td>
</tr>
<tr>
<td>Gaussian noise (Var=0.001)</td>
<td>0.9683</td>
<td>0.8430</td>
<td>0.4505</td>
<td>0.9829</td>
</tr>
<tr>
<td>Salt and pepper (density=0.001)</td>
<td>0.9976</td>
<td>0.98</td>
<td>0.9472</td>
<td>0.9853</td>
</tr>
<tr>
<td>Speckle noise (density=0.001)</td>
<td>NA</td>
<td>NA</td>
<td>0.9472</td>
<td>0.9865</td>
</tr>
<tr>
<td>Scaling (40%)</td>
<td>0.9866</td>
<td>0.8394</td>
<td>NA</td>
<td>0.9041</td>
</tr>
<tr>
<td>Cropping (50%)</td>
<td>0.7244</td>
<td>0.7240</td>
<td>NA</td>
<td>0.7835</td>
</tr>
<tr>
<td>JPEG (80%)</td>
<td>1.000</td>
<td>0.9870</td>
<td>NA</td>
<td>0.9850</td>
</tr>
<tr>
<td>JPEG (50%)</td>
<td>1.000</td>
<td>0.8670</td>
<td>0.8956</td>
<td>0.9838</td>
</tr>
<tr>
<td>Histogram equalization</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>0.9812</td>
</tr>
<tr>
<td>Image sharpening</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>0.9861</td>
</tr>
</tbody>
</table>
the other hand to improve the throughput of watermarking algorithm, the power of watermark is determined with a two-input fuzzy system. Experimental results demonstrate that the proposed scheme is robust against common non-geometric and geometric attacks.

REFERENCES


Figure 12: Some used images for evaluating the robustness of algorithm against various attacks

