A FRACTAL WATERMARKING SCHEME FOR IMAGE IN DWT DOMAIN

ABSTRACT

A new digital watermark approach based on the fractal technology in the Disperse Wavelet Transform domain is proposed in this paper. Firstly, we constructed the domain subtrees and the range subtrees in the wavelet decomposed subbands. Secondly, searched for the self-similar domain subtrees by the fractal transforms. The selected domain subtrees were regarded as the watermark embedded trees. Then, the watermarking signals were embedded, detected or extracted according to the JND model. Finally, A large number of attacking experiments results test security and robustness of the method.

Key words: Disperse Wavelet Transform; the domain subtree; the Fractal Transform.

I. INTRODUCTION

With the rapid development of information technology, Images are becoming more and more popular in the internet. There is an urgent need for copyright protection of images against pirating. Therefore, image watermark or signature algorithms represent a new, fast emerging area in image processing.

After researching on many images, we find that there exists a certain self-similarity among the different parts of an image or between the parts and the whole of an image. This self-similarity is an inherent attribute of the image. That is to say, which is hard to be changed and removed unless the quality of the image isn't guaranteed [1].

The wavelet transform coefficients of an image have good similarities which represent the self-similarity of the image. Because one character of the discrete wavelet decompose of an image is that it keeps the spatial local character of the image. Obviously, this kind of self-similarity can combine with the fractal method very well [2].

This paper presents a novel watermarking approach based on the fractal theory and the discrete wavelet transform. In section II, we propose the watermark embedding method based on the fractal transform in the DWT Domain. In section III, we discuss the watermark detecting (or extracting) process of the proposed scheme. At last, many experimental results represent the excellence of the scheme in section IV.

II. THE WATERMARK EMBEDDING METHOD BASED ON THE FRACTAL TRANSFORM

The watermark embedding scheme contains the following steps:

Step 1. Processing the watermark signals

Firstly, we process the watermark, then calculate the maximum watermark capacity according to the character of the image so as to guarantee the energy of the embedded watermark signals within the permitted range.

Supposing the watermark after preprocessing is \( W'_w = \{w'_i | w'_i \in \{0,1\}, i = 1,2,\cdots,N\} \), taking the later embedding into account, we change the watermark into:

\[
T : w_i^0 \rightarrow w_i = \begin{cases} 
1, & \text{if } w_i^0 = 1 \\
-1, & \text{if } w_i^0 = 0 
\end{cases}
\]

Finally, we get the information data which are going to be embedded into the image:

\[
W = \{w_i | w_i \in \{-1,1\}, i = 1,2,\cdots,N\} \quad (1)
\]

Step 2. Transforming the original image by the DWT and constructing the sets of domain subtrees and range subtrees which we called in this paper, then searching the self-similar domain subbands which the watermark data are embedded into.

The DWT is identical to a hierarchical subband system, where the subbands are logarithmically spaced in frequency and represent an octave-band decomposition. By the DWT, we have three parts of multiresolution representation (MRR) and a part of multiresolution approximation (MRA). The subbands in the 1st level labeled LH_1, HL_1 and HH_1 of the MRR represent the finest scale wavelet coefficients. To obtain the next coarser scale of wavelet coefficients, the subband LL_1 (that is, MRA) is further decomposed and critically subsampled… and so on. An image is decomposed into 3L+1 subbands for L scales.

As the construction approach of domain blocks and range blocks by the traditional fractal coding in the spatial domain, we select wavelet transform coefficients from the L level subband to the 1st level subband to construct the domain subtree D and construct the range subtree R from the L-1 level subband to the 1st level subband: Each wavelet coefficient in the L level subband has the corresponding relation with 2x2 coefficients in the L-1 level subband, 4x4 coefficients in the L-2 level subband …… and \((2^{L-1})\times(2^{L-1})\) coefficients in 1st level subband. Therefore, every coefficient of the L level subband on the vertical, horizontal and diagonal directions with its corresponding coefficient group can construct the domain tree D whose size is \(4^{L-1}\). In the same way, every wavelet coefficient in the L-1 level subband with its corresponding coefficient group can construct the range tree R. For example, if the root joint of the wavelet tree from the third level of the wavelet decompose together with its coefficient group can form a wavelet tree R whose size is 63. Obviously, if the size
of the original image is 512×512, then there will be 1024 domain subtrees D and 4096 range subtrees R after the 4 levels decomposition. These works are similar to construct non-overlapping image blocks of the fractal transform in the spatial domain[3].

According to this approach, the image block matching problem by the fractal transform in the spatial domain has been changed into the domain subtree matching problem in the wavelet domain. That is to say, we search those wavelet domain subtrees which are best-matched to their range subtrees R after the proper fractal transform, and then we call them self-similar domain subtrees.

Here, the correspondent spatial blocks of the domain subtrees are no overlapping.

Step 3. determining the position and the quantity of the embedding tree.

In step 2, we constructed the domain subtrees D and the range subtrees R, we can use the fractal transform in the DWT domain to search for the domain subtree D which is self-similar, and regard it as the embedding tree that we can embed the watermark into. The calculation method of the self-similarity in the DWT domain is different from the affine transform in the spatial domain, the fractal affine transforms of the DWT domain is τ(D)→R, which are composed of the geometry transform, the combination transform and the coefficient transform[4].

When we search for the best-matched subtree D, every D has a similar error which is the minimum between it and its corresponding range R by the affine transform.

Definition: Let the error threshold be ε₀, and for the domain subtree Dᵢ, if there is a range subtree Rᵢ which can make

\[ \min(d(Rᵢ, τ(Dᵢ))) \leq ε₀ \]  

then we call Dᵢ the self-similar domain subtree which can be embedding the watermark into.

Obviously, the error threshold ε₀ is very important here, it determines the attribute and classification of the domain subtree D. We can divide the domain subtrees into two kind of the embedding subtree and the non-embedding subtree, and each D only belongs to one of them.

Step 4. Embedding the watermark into the embedding subtree using the JND model in the DWT domain.

After step 3, we can get the embedded domain subtree \( D^{D} \) (i=1,2,…,M), we will embed the watermark in the formula (1) into these domain subtrees.

According to the JND model [5]:

\[
JND(x, y) = T(l, s, x, y) = \text{frequency}(l, s) \times \text{minance}(l, x, y) \times \text{texture}(l, x, y)^{0.04}
\]

(3)

we can calculate JND(x,y) of every point in the embedding tree D, then embed the watermark into it:

\[
f^{w}(x, y) = \begin{cases} 
  f(x, y) + JND(x, y) \times w_{x,y} & \text{if } f(x, y) > JND(x, y) \\
  f(x, y) & \text{otherwise}
\end{cases}
\]

(4)

Where JND(x,y) is the JND value of the point (x,y) in the embedding subtree D, f(x,y) is the coefficient of subtree D, \( w_{x,y} \) is the watermark which is embedded into the coefficient f(x,y), and \( f^{w}(x,y) \) is the coefficient of the domain subtree D after embedding \( w_{x,y} \).

Step 5. we transform the IDWT to the image, then we get the watermarked image.

### III. THE WATERMARK DETECTING (OR EXTRACTING) METHOD BASED ON THE FRACTAL TRANSFORM

The detecting (extracting) process of the watermark is the reverse process of the above embedding process, the steps are as follows:

Step 1. Take the wavelet transform on the original image, construct the domain subtrees and the range subtrees according to the method in section II, and then according to the threshold (secret key) ε₀ to determine the positions and the quantities of the embedded subtrees, and calculate the wavelet coefficient f(x,y) and the JND(x,y) of the corresponding point (x,y);

Step 2. We decompose the authentication image with the same wavelet transform, we can construct the domain subtrees and the range subtrees according to the method in step 1, we also can get the same position and quantity of the domain subtrees as well as the coefficient \( f^{w}(x,y) \) in the authentication image.

Step 3. The formula that the watermark is extracted from the domain subtree is as follows:

\[
 w^{*}_{x,y} = f^{*}(x, y) - f(x, y)
\]

\[
w^{*}_{x,y} = \frac{w^{*}_{x,y}}{JND(x, y)}
\]

Step 4. Processing the watermark

We can take the different process according to the different kind of the embedded watermark. If the embedded watermark is no-meaning as two-value stochastic sequence, we can judge it by the correlative detection method, that is depending on the following formula:

\[
\rho = \frac{w \cdot w^{*}}{\sqrt{E_{w} \cdot E_{w^{*}}}}
\]

(5)

Calculating the \( \rho \), and comparing it with the pre-set threshold \( \rho_{0} \) we can confirm whether there is the watermark in the authentication image.
If the embedded watermark like image or words has some meaning, we can process it on the reverse pre-process, so we can get the image watermark or the words information.

IV. THE EXPERIMENTAL RESULTS

A. The stochastic sequence watermark tests

The watermark that we used is binary stochastic sequence, i.e. -1 or 1, its length is 1500, and we have done the test with the 512×512×8 standard test image(Figure 1). We transformed the image by the 4 levels DWT, and constructed corresponding 4096 range subtrees R and 1024 domain subtrees D, the size of R is 63, the size of D is 255. Here the method we searched for D like the method in section II. For every D, we embedded the watermark into the 15 coefficients, and the number of the embedding D is 100. Then we embedded the watermark into these domain subtrees D, and got the watermarked image(Fig 2). Here we let \( \rho \) be 0.5 and consequently calculated the detection results of the image by the method in section III.

![Figure 1. Standard test image](image1)

![Figure 2. Watermarked image](image2)

**Test 1. JPEG compression**

We carried out JPEG compression to the watermarked image above (Fig 2), then detected the watermark according to the method in section III, and correspondingly received the following test results.

<table>
<thead>
<tr>
<th>QF</th>
<th>Correlative rate ( \rho )</th>
<th>Watermark or no</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>0.95</td>
<td>YES</td>
</tr>
<tr>
<td>60</td>
<td>0.87</td>
<td>YES</td>
</tr>
<tr>
<td>40</td>
<td>0.76</td>
<td>YES</td>
</tr>
<tr>
<td>20</td>
<td>0.44</td>
<td>NO</td>
</tr>
</tbody>
</table>

We used QF (Quality Factor) to indicate the effect of JPEG compression. The bigger the QF is, the better the compression quality will be. JPEG compression considers of the characteristic of Human Visual System. This is as the same as what we think while we embedded watermark. Therefore our method is robust.

**Test 2. Cropping**

We cropped the watermarked image above on the different proportions. Starting from the left up corner, and the cropping zone is square (cropping proportion = the width of cropping zone/the width of the embedded image), finally we got corresponding test results:

<table>
<thead>
<tr>
<th>Cropping proportion</th>
<th>Correlative rate ( \rho )</th>
<th>Watermark or no</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/32</td>
<td>0.90</td>
<td>YES</td>
</tr>
<tr>
<td>1/16</td>
<td>0.82</td>
<td>YES</td>
</tr>
<tr>
<td>1/8</td>
<td>0.69</td>
<td>YES</td>
</tr>
<tr>
<td>1/4</td>
<td>0.47</td>
<td>NO</td>
</tr>
</tbody>
</table>

From table 2 above we can see, when the cropping proportion is smaller than 25%, the correlative rate of the watermark is very high. Of course, with the cropping proportion gradually increasing, the correlative rate is declining quickly. This is mainly because that the cropping destroys the local similarity of an image. But at this time, the visual effect of the image also declined much. To attacker, the attack has little significance.

**Test 3. Median filtering**

We filtered the watermarked image by different median filters, then detected it, and the results are as follows:

<table>
<thead>
<tr>
<th>Median filter</th>
<th>Correlative rate ( \rho )</th>
<th>Watermark or no</th>
</tr>
</thead>
<tbody>
<tr>
<td>3×1</td>
<td>0.92</td>
<td>YES</td>
</tr>
<tr>
<td>3×3</td>
<td>0.80</td>
<td>YES</td>
</tr>
<tr>
<td>5×1</td>
<td>0.74</td>
<td>YES</td>
</tr>
<tr>
<td>5×5</td>
<td>0.61</td>
<td>YES</td>
</tr>
</tbody>
</table>

Median filtering brings many losses of the details of the image, but what it destroys most is the high frequency coefficients. The coefficients we chose while embedding were mostly in the low frequency zone, so this method is robust to median filter.
**Test 4. Adding noises**

We added Gauss noises of the four different intensity levels to the embedded image, and calculated the correlative rate, the results are as follows:

<table>
<thead>
<tr>
<th>Noise variance</th>
<th>Correlative rate $\rho$</th>
<th>Watermark or no</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.91</td>
<td>YES</td>
</tr>
<tr>
<td>4</td>
<td>0.83</td>
<td>YES</td>
</tr>
<tr>
<td>6</td>
<td>0.66</td>
<td>YES</td>
</tr>
<tr>
<td>8</td>
<td>0.45</td>
<td>NO</td>
</tr>
</tbody>
</table>

The destruction of noise to the image is whole. With the noise intensity adding, the correlative rate decline gradually. The key reason is that the image similarity is destroyed and the wavelet coefficients are changed, but now the image visual quality declined obviously.

**B. Image watermark tests**

We used 256×256×8 standard test image Lena.bmp(Fig 3) as the original image, and the watermark we used is the two-value image of 32×32 Sy.bmp (Fig 4). To ensure the safety after embedding, we did the chaos processing before embedding and embedded the watermark according to the method in section II and got the watermarked image (Fig 5).

To imitate the damage condition of the image attacking, we used the common image processing like tests above to attack the embedded image and then extracted the watermark according to the method in section III, the results are as follows:

**Test 1. JPEG compression**

We carried out JPEG compression to the watermarked image above (Fig 5), then extracted watermark according to the method in section III. The results are shown in the following table:

<table>
<thead>
<tr>
<th>QF</th>
<th>The Extracted watermark</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td></td>
</tr>
</tbody>
</table>

**Test 2. cropping**

We cropped the watermarked image above on the different proportions as Test 2 in section IV.A, then extracted watermark. The results are shown in the following table:

<table>
<thead>
<tr>
<th>Cropping proportion</th>
<th>The Extracted watermark</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/32</td>
<td></td>
</tr>
<tr>
<td>1/16</td>
<td></td>
</tr>
</tbody>
</table>

**Test 3. Median filtering**

We filtered the watermarked image in different median filters, then extracted the watermark. The results are as follows:

<table>
<thead>
<tr>
<th>Median filter</th>
<th>The Extracted watermark</th>
</tr>
</thead>
</table>
Test 4. Adding noise

We did the similar tests as the same as the test 4 in section IV.A. The results are shown in the following table:

<table>
<thead>
<tr>
<th>Noise variance</th>
<th>Extracted watermark</th>
<th>Noise variance</th>
<th>Extracted watermark</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>水印</td>
<td>6</td>
<td>水印</td>
</tr>
<tr>
<td>4</td>
<td>水印</td>
<td>8</td>
<td>水印</td>
</tr>
</tbody>
</table>

V. Conclusions

The scheme combines the fractal transform and the DWT, and adopted the image similarity property to search the domain subtree that the watermark is embedded into. The simulation results show that the proposed method can actually survive the general image processing attacks and can be presented to claim the invisibility and robustness.

References