Robust Fractal Watermarking on Continue-tone Images

Ping-Sung Liao, Chun-Chi Chen, Jeng-Shyang Pan
Cheng-Shiu University, National Kaohsiung University of Applied Sciences

Abstract

In this study, a novel fractal watermarking is proposed to enforce the robustness of data hiding against possible attack. The proposed fractal watermarking is mixed of three different approaches: inner/exterior search space, odd/even search space, and two classes of isometric search space. During watermark extraction, the watermark bit is always determined by the result of a voting mechanism on the spatial coordinate with respect to three search spaces mentioned above. Experimental result shows the proposed can react against the attack JPEG compression.

Keywords: robust fractal watermarking, data hiding, voting mechanism.

1. Introduction

In recent years, computers are broadly and deeply linked to many different information processing applications that can be easily accessed from any place around the world. As a result, to protect the copyright of digital contents has become an urgent discipline. In the latter half of the 1990s, digital watermarking has been gained popularity as a research topic to embed a security message into digital documents that may be one of images, music, audio clip and video clip. Based on the characteristics of digital contents, many kinds of digital watermarking techniques based on spatial domain or DCT transform [1] have been proposed by researchers to protect the copyright of the diversified contents. In 1996, Puate and Jordan proposed the first fractal watermarking whose concept was rooted on applying two different search regions (inner and exterior regions) to embed watermark bit (one and zero) [2]. In 2000, Li and Wang utilized the isometric property to embed the watermark bit (one and zero) by two kinds of isometric plane [3]. Since then, there were few papers to investigate the digital water marking based on fractal coding. In this study, authors proposed a robust approach of fractal watermarking that includes odd-even search except both popular searches: local search restriction (LSR) and odd-even search.

2. Fractal image coding

The fractal theory (especially the analysis of fractal dimension) has been successfully applied in various applications of pattern recognition. In the application of image rendering, any interesting scene is easily accomplished by taking specific affine transformations recursively. For example, Spleenwort fern, Dragon, Sierpinsky triangular, Maple leaf and the Mandelbrot set in a landscape are the popular images derived by different affine transformations. In the past, fractal theory investigated by Mandelbrot enable us to view fractals as amazing tools to create strange objects from either determined approaches or random approaches [4]. Besides, the critical step to model real world objects was the concept of Partitioned Iterated Function Systems (PIFS) proposed by Barnsley. The PIFS theory assumes that one small region of an image may be approximated by other self-similarity region in the same image by the use of the IFS simple transformation [2]. In other words, let original image be T and the recovered image be R,

\[ R = \bigcup_{m=1}^{M} \bigcup_{n=1}^{N} (P_m \circ \omega_n(P_m)), \]

where \( P_m \) is a part of image T, \( \omega_n(P_m) \) is the affine transformation of \( P_m \). [5]

In order to minimize the difference between the range block \( r \) (destination) and the one suitable domain block \( D_i \) (source) under the affine transformation given in equation (2), PIFS will fully search the domain pool to get the best one with minimum mean square error. After fractal coding, the result relative to each range block is position parameter \( (x, y) \), scaling factor \( a \), brightness shift \( \beta \) and isometry \( \iota \).

\[ r = \alpha \cdot D + \beta \]

Because the time used for fractal encoding and fractal decoding is not symmetric, and the former always consumes much time to realize the encoding work, local PIFS (LPIFS) was developed to speed up the computation of fractal encoding. For many nature images without frequently abrupt change, the LPIFS result seems perceptible when we look in front of the fractal decoded image.
3. Literature Review

To sign a watermark signature into an image, Puade and Jordan adopted LPIFS to perform digital watermarking. For example, an image T is chosen to embed a black/white image (logo). If a watermark bit \( w_i \) to be one will be embedded in the range block \( R_b \), the search space will be the exterior region \( \text{LSR}_1 \) around the range block as shown in Figure 1. On the contrary, if a watermark bit \( w_i \) to be zero will be embedded in the range block \( R_b \), the search space will be the inner region \( \text{LSR}_0 \) around the range block. Obviously, one range block is used to embed one watermark bit.

This fractal watermarking consists of coding and decoding procedure. The resulting image by fractal decoding is finally considered as the watermarked image [2]. Lifting the restriction of search space, the procedure of watermark extraction is conducted the same as that of watermark embedding is done. First, LPIFS is performed over the union region of \( \text{LSR}_0 \) and \( \text{LSR}_1 \). Undoubtedly, the best domain block would be either \( \text{LSR}_0 \) or \( \text{LSR}_1 \). If it position is in \( \text{LSR}_1 \), the watermark bit \( w_i \) is one. Otherwise, the watermark bit \( w_i \) is zero.

There are eight kinds of isometry for each affine transformation [3, 4]. These isometric transformation are (1) identity: \( \iota_0 \), (2) rotation around center through +90°: \( \iota_1 \), (3) rotation around center through +180°: \( \iota_2 \), (4) rotation around center through +270°: \( \iota_3 \), (5) orthogonal reflection about the vertical axis: \( \iota_4 \), (6) orthogonal reflection about the vertical axis, and rotation around center through +90°: \( \iota_5 \), (7) orthogonal reflection about the vertical axis, and rotation around center through +180°: \( \iota_6 \), and (8) orthogonal reflection about the vertical axis, and rotation around center through +270°: \( \iota_7 \).

During the watermark embedding, eight isometric transformations are classified into two groups. One is the union of odd isometries: \( \iota_1, \iota_3, \iota_5, \) and \( \iota_7 \), the other is the union of even isometries: \( \iota_0, \iota_2, \iota_4, \) and \( \iota_6 \). If a watermark bit \( w_i \) to be one will be embedded in the range block \( R_b \), the search space will be the neighboring region that is around the range block and possesses odd isometries, as shown in Figure 2. On the contrary, if a watermark bit \( w_i \) to be zero will be embedded in the range block \( R_b \), the search space will be the neighboring region that is around the range block and possesses even isometries.

During watermark extraction, applying the fractal encoding on the watermarked image will result in a set of fractal parameters. The isometry parameter would reflect what the watermark bit is because the watermarked image is the approximated image derived from taking the fractal compression and digital watermarking together on the original image.

Fig. 1 The search space proposed by Puade and Jordan

Fig. 2 The search space proposed Li and Wang

3. Robust Fractal Watermarking

In this study, authors proposed a new search strategy called Odd-Even partition to do digital watermarking. This partition as shown in Figure 3 seems like the chessboard. Around range block \( R_b \) positioned in the center, the collection of gray domain blocks is the even partition and the other collection of white domain blocks is the odd partition. They will be used to embed watermark bit into a range block. When a watermark bit is zero, fractal coding is performed only at the even partition to seek the
best set of fractal parameters. Oppositely, if a watermark bit is one, fractal coding is run only at the odd partition to seek the best set of fractal parameters.

So far we have discussed three different search strategies based on fractal image coding scheme for digital watermarking. Next, author aimed to mix them together for the sake of more robust watermark extraction under the bad conditions such as JPEG compression. Figure 4 illustrates the mixture of these three search strategies. Now, we define two special search spaces for digital watermarking. The first group $G_0$ points to the domain blocks with three indexes LSR_0, even isometries and even partition, respectively. The second group $G_1$ points to the domain blocks with indexes LSR_1, odd isometries and odd partition, respectively. Our approach about robust fractal watermarking is described below.

**Watermark embedding process**

**STEP 1**: Partition an image into several range blocks

Some range blocks are randomly chosen to embed watermark message by a random seed $s$.

**STEP 2**: Permute the watermark

Before the watermark is embedded into an image, the watermark message is permuted by a random seed (key $κ$), which is the key to decry the sequence of extracted watermark bits into their original sequence.

**STEP 3**: Perform fractal coding for data embedding

If a watermark bit $w_i$ is 1, only the search space $G_1$ will be considered for fractal watermarking. Otherwise, if a watermark bit $w_i$ is 0, only the search space $G_0$ will be chosen for fractal watermarking.

**Watermark extraction process**

**STEP 1**: Extract watermark from the watermarked image

The random seed $s$ is applied to filter out the range blocks for digital watermarking. For the $j$-th range block $R_b(j)$, taking a full search over its neighboring domain pool will result in three indexes $LSR(j)$, $ι(j)$ and $ODD-EVEN(j)$, respectively. According to Table 1, the watermark bit relate to the $j$-th range block will be determined. Apparently, the final result is decided by the side of many votes.

**STEP 2**: Rearrange the extracted watermark bits

An inverse permutation with decryption key $κ$ is necessary to recover the extracted watermark bits into their initial states.

**Table 1 Voting mechanism for the determination of watermarking bit**

<table>
<thead>
<tr>
<th>$LSR(j)$</th>
<th>$ι(j)$</th>
<th>$ODD-EVEN(j)$</th>
<th>$w_i(j)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

4. Experiments and Discussions

To demonstrate the performance of the proposed method, test image ‘Lena’ with size 256 x256 as shown in Fig. 5 is
used for evaluation. The watermark with 32 x 32 pixels is given in Fig. 6. For fractal image coding, the size of range block size is 4x4. The watermarked images based on LSR search, odd-even search, isometry search and mixed search are shown in Fig. 7, Fig. 9, Fig. 11 and Fig. 13. It is difficult to perceive a subtle difference among these watermarked images. Their corresponding extracted watermark are shown in Fig. 8, Fig. 10, Fig. 12 and Fig. 14 without the attack of JPEG compression. Their bit correction rates (BCR) are near 100%. The result of Table 2 shows that even at the worse case of JPEG compression (JPEG quality =60%) the BCR of mixed search is greater than 78%.

5. Conclusions

A novel watermarking with voting mechanism and data hiding based on fractal image coding scheme is proposed here. During the process of fractal watermarking, the core concern is robustness, which can tolerate the attack of JPEG compression. Experimental results show that the quality of the watermarked images by the proposed method is almost perceptively satisfactory and voting mechanism is workable to overcome the attack of JPEG attack. In the future study, a detail investigation will be proceeded to strongly prove that such a robust fractal watermarking are able to protect the copyright and the ownership of any images.

References


Fig. 5 Lenna image
Fig. 7 Watermarked lenna image by LSR search

Fig. 6 Watermark
Fig. 8 Extracted watermark from Fig. 7
Fig. 9 Watermarked lenna image by odd-even search

Fig. 11 Watermarked lenna image by isometry search

Fig. 10 Extracted watermark from Fig. 9

Fig. 12 Extracted watermark from Fig. 11

Fig. 13 Watermarked lenna image by mixed search

Fig. 14 Extracted watermark from Fig. 13
Table 2  BCR ratio (%) for different search during fractal watermarking

<table>
<thead>
<tr>
<th>JPEG quality</th>
<th>LSR search</th>
<th>Odd-even search</th>
<th>Isometry search</th>
<th>Mixed search (LSR + odd-even + isometry)</th>
</tr>
</thead>
<tbody>
<tr>
<td>60%</td>
<td>79.00</td>
<td>64.55</td>
<td>78.12</td>
<td>77.83</td>
</tr>
<tr>
<td>70%</td>
<td>82.52</td>
<td>64.55</td>
<td>82.91</td>
<td>81.05</td>
</tr>
<tr>
<td>80%</td>
<td>88.86</td>
<td>73.44</td>
<td>88.48</td>
<td>88.18</td>
</tr>
<tr>
<td>90%</td>
<td>94.34</td>
<td>81.54</td>
<td>94.73</td>
<td>94.34</td>
</tr>
<tr>
<td>100%</td>
<td>99.61</td>
<td>99.12</td>
<td>100</td>
<td>99.61</td>
</tr>
</tbody>
</table>