A NOVEL CLASSIFICATION METHOD FOR IMAGE CODING

Ming-Sheng Wu

Dept. of Electrical Engineering, Cheng Shiu University, Kaohsiung, Taiwan
E-mail: sheng@mail.csu.edu.tw

ABSTRACT

In this paper, a novel classification method is proposed to speedup the fractal encoding speed. By using two DCT coefficients: lowest horizontal coefficient and lowest vertical coefficient, all of the image blocks including domain blocks and range blocks are classified into three classes: smooth type, horizontal/vertical edge type and diagonal/sub-diagonal edge type. For each range block to be encoded, only the MSE value between it and the domain block which the type is the same as it is calculated, since they have same texture. The experimental results show that the encoding speed of proposed classification method is about 3 times fast than that of the full searching method and the quality of retrieved image is almost the same.

1. INTRODUCTION

Fractal image compression is a great potential coding scheme since it has high compression ratio and good retrieved image quality. Its original idea was proposed by Barnsley [1-3] according to the iterated function system (IFS). In 1992, the first fractal image coding algorithm was realized by Jacquin [4]. The underlying premise of the coding scheme is based on the partitioned iteration function system (PIFS) which utilized the self-similarity characteristic in a nature image to achieve the purpose of compression.

The drawback of the fractal image compression is that the encoding process is time-consuming very much since, for each range block, a large amount of computations of similarity measure must be done to find the best matched domain block. Hence the main research topic of the field is focused on how to reduce the encoding time. At present, the main techniques used to speedup the encoder are utilizing partition and classification algorithms to decrease the computations of similarity measure [5-7]. Besides, the idea of the spatial correlation was also added in the fractal encoding algorithm to reduce the search space of the optimal solution. Truong et al. [8] limit the search space for the current range block on the neighborhoods of the matched domain blocks of the neighboring range blocks by utilizing the spatial correlations between neighboring blocks in both the domain pool and the range pool. Comparing with the full search algorithm, the algorithm achieves 2.6 times speedup ratio and obtains almost same retrieved image quality. Furao and Hasegawa [9] present a no search fractal encoding algorithm. In this algorithm, they select the domain block which has the same center with the range block to be encoded as the matched one, since the two blocks usually have the same texture in a nature image. Utilizing such characteristic and merging quad-tree technique, the authors achieve the fastest encoding velocity up to now, but both of the retrieved image quality and the compression ratio will be sacrificed.

In this paper, a novel classification method is proposed to speedup the fractal encoding speed. First by using the new classification method, all of the image blocks including domain blocks and range blocks will be classified as three classes: smooth type, diagonal/sub-diagonal edge type and horizontal/vertical edge type. Two DCT coefficients: the lowest horizontal coefficient $F(0,1)$ and the lowest vertical coefficient $F(1,0)$ are used to determine that the block is belonging to which type. Then for each range block to be encoded, only the MSE value between it and the domain block which the type is the same as it is calculated. The main reason of the search space in the same type domain pool is that the best matched solution usually exists in the space, since they have same texture. In other words, if the type of a domain block is not the same as the one of range block to be encoded, their MSE value need not to be calculated. We can find that by using the proposed classification method, the encoding speed is about 3 times fast than the full searching method and the quality of retrieved image is almost the same.

The rest of the paper is organized as follows. We introduce the conventional fractal image coding scheme in Section 2. Section 3 describes the proposed classification method in this paper. Some experimental simulations are performed in Section 4 to verify the improvement of our proposed method. Finally, a conclusion is made in Section 5.

2. FRACTAL IMAGE ENCODING

The fractal image compression is based on the local self-similarity property and PIFS. The related implementation process in practice is stated as follows.

Suppose the original gray level image $f$ is of size $m \times n$. To simplify the discussion, let $m$ be 256. The domain pool is defined as the set of all possible blocks
of size 16×16 of the image, which makes up
(256-16+1)×(256-16+1) = 58081 blocks. The
range pool is defined to be the set of all non-overlapping
blocks of size 8×8 of the image f, which makes up
(256/8)×(256/8)=1024 blocks. For each block v
from the range pool, the fractal transformation is
constructed by searching all of the elements in the
domain pool to find the most similar block. Let u
denote a sub-sampled domain block which is of the
same size as v. The similarity of u and v is measured
using mean square error (MSE), which is defined by

\[ MSE = \frac{1}{64} \sum_{j=0}^{7} \sum_{i=0}^{7} (u(i, j) - v(i, j))^2 \]  (1)

The fractal transformation allows the Dihedral
transforms of the domain blocks, i.e., the 8 orientations
of the blocks generated by rotating the blocks
clockwise at angles 0°, 90°, 180° and 270° and
flipping with respect to the line \( y = x \), respectively, as
shown in Table 1. Thus for a given block from the range
pool, there are 58081×8 = 464,648 MSE computations to
obtain the most similar block from the domain pool.
Thus, in total, one needs 1024×464,648 = 475,799,552
MSE computations to encode the whole image using this
full search compression method.

<table>
<thead>
<tr>
<th>Rotate 0°</th>
<th>Rotate 90°</th>
<th>Rotate 180°</th>
<th>Rotate 270°</th>
</tr>
</thead>
<tbody>
<tr>
<td>([0 \ 1])</td>
<td>([0 \ 1])</td>
<td>([-1 \ 0])</td>
<td>([0 \ -1])</td>
</tr>
<tr>
<td>([1 \ 0])</td>
<td>([-1 \ 0])</td>
<td>([0 \ -1])</td>
<td>([1 \ 0])</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flip with the line ( y=x ) from above</th>
</tr>
</thead>
<tbody>
<tr>
<td>([0 \ 1])</td>
</tr>
<tr>
<td>([1 \ 0])</td>
</tr>
</tbody>
</table>

The fractal transformation also allows the adjustment of
the contrast p and the brightness q on the
sub-sample block u. Thus the similarity is to minimize the quantity
\( d = \| p \cdot u + q - v \| \) , where \( u_k \), \( 0 \leq k \leq 7 \)
are the 8 orientations of u. By calculus, p and q can be
computed directly by

\[
p = \frac{\sum_{i,j} u(i,j) \cdot v(i,j) - \sum_{i,j} u(i,j)}{\sum_{i,j} v(i,j)} ,
\]

\[
q = \frac{\sum_{i,j} v(i,j) - \sum_{i,j} u(i,j)}{\sum_{i,j} v(i,j)} ,
\]

where \( N \) is the number of pixels of the range block and
\( \vec{1} = [1 \ 1 \ \cdots \ 1] \).

Finally, the position x and y of the domain block,
the contrast p, the brightness q, and the orientation k
constitute the fractal code of the given range block v.

For 256×256 image, both x and y require 8 and 8 bits,
respectively, to represent the position of the domain block. For contrast p, brightness q, and the orientation
k, 5, 7 and 3 bits are required, respectively. Hence one
needs 8×8×5+7+3=31 bits in total to encode a range
block. Finally, as v runs over all 1024 range blocks in
the range pool, the encoding process is completed.

To decode, one first makes up the 1024 affine
transformations from the compression codes. Next, one
chooses any initial image and performs the 1024 affine
transformations on the image to obtain a new image.
The transformation is proceeded recursively according to
contraction mapping theorem and Collage theorem
until the sequence of images converge to encoded image.
The stopping criterion of the recursion is designed
according to user’s application and the final image is the
retrieved image of fractal coding.

3. A NOVEL CLASSIFICATION METHOD FOR
IMAGE CODING

In this section, a novel classification method
proposed by us will be introduced. The main goal of
the method is to decrease the number of the MSE
computations at searching the best matched solution in
order to reduce the encoding time of fractal image. All
of the image blocks including range blocks and domain
blocks will be classified into three types: smooth type (S
type), horizontal/vertical type (H type), and
diagonal/sub-diagonal type (D type) according to their
texture by using two DCT coefficients: the lowest
horizontal coefficient \( F_0 \) and the lowest vertical
coefficient \( F_1 \). Two threshold angles \( T_{S1} \) and \( T_{S2} \)
and one threshold circle \( T_C \) are taken to determine the
threshold between different categories. Below, the
classification method for the image blocks will be stated
in detailed.

Assume f is a given image block of size 8×8, we
can compute \( F(1,0) \) and \( F(0,1) \) from

\[
F(1,0) = \sqrt{\frac{2}{8}} \sum_{j=0}^{7} \sum_{i=0}^{7} f(i,j) \cos \theta_i ,
\]

\[
F(0,1) = \sqrt{\frac{2}{8}} \sum_{j=0}^{7} \sum_{i=0}^{7} f(i,j) \cos \theta_j ,
\]

where \( \theta_i = (2j+1)/16 \), \( i = 0, 1, \cdots , 7 \).

The magnitude of \( F(1,0) \) reflects the intensity
variation between the upper half and lower half of image
block f and the magnitude of \( F(0,1) \) reflects the
intensity variation between the left half and right half of
image block f. For an image block f, if both \( |F(1,0)| \)
and \( |F(0,1)| \) are small, it means that the block f has not
clear texture. Hence the landscape on the block f is
tending to smooth. The block f will be classified as
smooth type and denoted as S type. Second, if \( |F(1, 0)| \) is large and \( |F(0, 1)| \) is small, \( f \) will have vertical edge. But if \( |F(1, 0)| \) is small and \( |F(0, 1)| \) is large, \( f \) will have horizontal edge. For the two cases, \( f \) is classified as horizontal/vertical type and denoted as H type. Finally, if both \( |F(1, 0)| \) and \( |F(0, 1)| \) are large, it means that \( f \) will have diagonal or sub-diagonal edge. Such \( f \) will be classified as diagonal/sub-diagonal type and denoted as D type.

A block is classified as H type if its position is located on between \( 0^\circ \) and \( T_{12}^a \) or between \( T_{13}^a \) and \( 90^\circ \). Once the H type blocks are determined, the S type blocks are also isolated from the rest ones composed of non-H type blocks. First, according to the original DCT absolute coefficient pairs \( \{F(1, 0), |F(0, 1)|\} \) which have not to be normalized, we find the image block which the distance is the most far with the origin of coordinate and its position is labeled as \( \|F(1, 0)\|_{\text{max}}, \|F(0, 1)\|_{\text{max}} \). Then the DCT absolute coefficient pair \( \{F(1, 0), |F(0, 1)|\} \) of all the rest blocks are divided by \( \left( \|F(1, 0)\|_{\text{max}}^2 + \|F(0, 1)\|_{\text{max}}^2 \right)^{1/2} \). By the regulative process, the positions of the rest image blocks are located in the unit circle between \( T_{11}^a \) and \( T_{22}^a \). The result is shown in Fig.2. If the position of an image block \( f \) is near the origin of coordinate, it will have smooth landscape since both \( |F(1, 0)| \) and \( |F(0, 1)| \) are small. Hence the image block \( f \) will be classified as S type. A circle which radius is \( T_c \) used to separate the S type blocks from the rest ones. A block is classified as S type if its position is located in the circle. Finally, the image blocks which are not belonging to H type and S type will be classified as D type. For these blocks, they have clear diagonal or sub-diagonal edge since their \( |F(1, 0)| \) and \( |F(0, 1)| \) are large.

To determine the type of all of the image blocks from the range pool and domain pool, the blocks is executed first the DCT transformation to find their \( F(1, 0) \) and \( F(0, 1) \). Then the classification algorithm is used to determine that these image blocks are belonging to which type according to their \( F(1, 0) \) and \( F(0, 1) \). First, H type blocks are separated from all of the image blocks. For each image block, we take the absolute values of \( F(1, 0) \) and \( F(0, 1) \) to constitute a DCT absolute coefficient pair \( \{F(1, 0), |F(0, 1)|\} \). After normalization, the normalized DCT absolute coefficient pair \( \{F(1, 0), |F(0, 1)|\}_N \) is obtained and is labeled on the Cartesian coordinate system in which \( F(1, 0) \) is x-coordinate and \( F(0, 1) \) is y-coordinate. The result is shown in Fig.1. We can find that the positions of all the image blocks normalized are located on the unit circle of first quadrant. According to the texture characteristic of image block, if DCT absolute coefficient pair of an image block \( f \) is near the point \((1, 0)\), it will have clear vertical edge since \( |F(1, 0)| \) is large and \( |F(0, 1)| \) is small. Hence the image block \( f \) will be classified as H type. For the same reason, if DCT absolute coefficient pair of an image block \( f \) is near the point \((0, 1)\), it will have clear horizontal edge since \( |F(1, 0)| \) is small and \( |F(0, 1)| \) is large. Such an image block \( f \) will also be classified as H type. Two threshold angles: \( T_{11}^a \) and \( T_{22}^a \) are used to separate H type blocks from non-H type ones.

The classification step in detail for the classification method is stated as follows:

1. For all of the image blocks including domain blocks and range blocks, calculate the \( F(1, 0) \) and \( F(0, 1) \) from (2) and (3), respectively, and constitute the DCT absolute coefficient pair \( \{F(1, 0), |F(0, 1)|\} \).
2. Normalize the DCT absolute coefficient pair \( \{F(1, 0), |F(0, 1)|\} \) of all the image blocks and label the position of each image block according to the normalized DCT absolute coefficient pair.
\[ F(1,0) \mid F(0,1) \] on the Cartesian coordinate system. The result is shown in Fig.1.

3. Determine the H type blocks from Fig.1. A block is classified as the type if its position is located on between \( 0^\circ \) and \( T_{i0} \) or between \( T_{i2} \) and \( 90^\circ \).

4. For the rest ones in Step 3, find the pair \( \{ F(1,0) \mid F(0,1) \} \) according to their un-normalized DCT absolute coefficient pair \( \{ F(1,0) \mid F(0,1) \} \) obtained in Step 1. Divide the \( \{ F(1,0) \mid F(0,1) \} \) of all the rest blocks by \( \{ F(1,0) \mid F(0,1) \} \). Label their position on the Cartesian coordinate system. The result is shown in Fig.2.

5. Determine the S type blocks from Fig.2. A block is classified as the type if its position is located in the circle which radius is \( T_c \).

6. Assign the rest blocks in Step 5 as D type.

Once the classification is completed, the encode process will be executed. Each range block is limited the search space of the best matched solution to the same type domain blocks in order to decrease the number of MSE computations and speedup the encoder.

4. EXPERIMENTAL RESULTS

The images Lena and Pepper are tested to demonstrate the encoding time and retrieved image quality of the proposed classification method in comparison to the full searching method. For a given image of size 256 x 256, the size of range block is chosen to be 8 x 8. The software simulation is done using BCB on a Celeron Core2 Duo 2.0GHz, 2G RAM, windows XP PC.

The values of both \( T_{i0} \) and \( T_{i2} \) are 12.5\(^\circ\) and 77.5\(^\circ\), respectively. The radius \( T_c \) is set to be 0.1. The tested image is Lena. Fig. 3 shows the results of the proposed classification method in comparison with the full searching method. Fig 3(a) is the original Lena image. 3(b) is the initial image used to retrieve the fractal-compressed image. 3(c) and 3(d) show the retrieved image using the full searching method and the proposed classification method, respectively. As listed, the encoding time of the proposed classification method is about one-third of the full searching method, since the MSE computations of the proposed classification method is about one-third of the full searching method. Moreover, the retrieved image quality for the two encode methods is almost the same. The experimental results of the Pepper are shown in Fig. 4. The threshold values: \( T_{i0} \), \( T_{i2} \), and \( T_c \) are the same as Lena. Similarly, the encoding time of the proposed classification method is about one-third of the full searching method, while the quality of the retrieved image is also almost the same. The penalty is 0.15 dB decay in PSNR.

5. CONCLUSION

In this paper, a classification method was proposed to speedup the fractal encoding speed. All of the image blocks including range blocks and domain blocks will be classified into three types: smooth type(S type), horizontal/vertical type(H type), and diagonal/sub-diagonal type(D type) according to their texture by using two DCT coefficients: the lowest horizontal coefficient \( F(1,0) \) and the lowest vertical coefficient \( F(0,1) \). Two threshold angles \( T_{i0} \) and \( T_{i2} \) and one threshold circle \( T_c \) are taken to determine that the block is belonging which category. Each range block is limited the search space of the best matched solution to the same type domain blocks in order to decrease the number of MSE computations and speedup the encoder.

Experimental results show that the encoding time of the proposed classification method is about one-third of the full searching method, since the MSE computations of the proposed classification method is about one-third of the full searching method. Moreover, the retrieved image quality for the two encode methods is almost the same and only little decay in PSNR.

REFERENCES

Fig 3. (a). Original image. Lena of size 256*256.
(b). The initial image for the decoder of the fractal compression.
(c). Full searching method, PSNR=28.91, Time=2577.90 sec.
(d). Proposed classification method, PSNR=28.82, Time=769.81 sec.
Fig 4. (a). Original image, Pepper of size $256 \times 256$.
(b). The initial image for the decoder of the fractal compression.
(c). Full searching method, PSNR=29.85, Time=2579.02 sec.
(d). Proposed classification method, PSNR=29.70, Time=783.61 sec.