

## Short Paper

---

# Direct Allocating the Dihedral Transformation for Fractal Image Compression

D. J. DUH, J. H. JENG\* AND S. Y. CHEN

*Department of Computer Science and Engineering  
Yuan Ze University  
Taoyuan, 320 Taiwan*

*\*Department of Information Engineering  
I-Shou University  
Kaohsiung County, 840 Taiwan*

Fractal image compression exploits the self-similarity of an image to achieve the purpose of compression. In the standard algorithm, eight Dihedral transformations are applied on domain blocks to increase the codebook size, and therefore, the quality of reconstructed image can be improved. However, such mechanism consumes approximately eight times of the encoding time. On the other hand, if no transformation is performed in order to speedup the encoder, the image quality will decay because the codebook is not large enough. In this paper, we propose a direct allocating method to predict the desired orientation and the similarity measure is performed on this orientation only. Simulations show that the encoding time is almost the same as that of the method without transformations while the image quality is close to that of the standard method.

**Keywords:** fractal image compression, PIFS, Dihedral transformation, DCT, AC coefficients

## 1. INTRODUCTION

The idea of the fractal image compression (FIC) is based on the assumption that the image redundancies can be efficiently exploited by means of block self-affine transformations [1]. The fractal transform for image compression was originally introduced by M. F. Barnsley and S. Demko [2]. The first practical fractal image compression scheme was proposed by A. E. Jacquin [3], E. W. Jacobs, Y. Fisher and R. D. Boss [4], which utilized block-based transformations and an exhaustive search strategy. Their approach was an improved version of the system patented by Barnsley [5]. The main disadvantage of fractal image compression is the computation complexity, *i.e.*, the encoding speed. However, it provides high compression ratio and fair image quality and thus attracts many off-line applications.

The intuitive concept of fractal coder is the search of the most similar block from a large codebook. The codebook consists of all possible blocks from the original image, in which each block is transformed using the Dihedral transformation to obtain eight orien-

---

Received March 3, 2005; revised May 17, 2005; accepted June 1, 2005.  
Communicated by Kuo-Chin Fan.

tations. The exhaust search together with these eight orientations constitutes the essence part of the computations. Commonly used methods to speedup the encoder are the reduction of the codebook size such as the partition or classification of image blocks [6, 7]. Another attempt is the simplified encoding method in which no Dihedral transformations are performed [8-10]. This method does increase the speed about eight times but it decrease the image quality because the lack of sufficient domain pool.

Jeng *et al.* [11, 12] used DCT and Hadamard transforms to speedup the fractal encoding process. The main contribution is to remove the redundant calculations of the Dihedral transformations when minimizing the MSE (mean square errors) in the frequency domain. No pre-classification method is applied to predict the best Dihedral transformation for the best matched domain block. Speedup factors of 2 to 4 are achieved without any loss of image quality. This is because all of the eight orientations are indeed calculated in their algorithm.

In this paper, we propose a direct allocating method to predict the most preferred orientation. The prediction is accomplished by comparing three DCT coefficients of the given range block and the domain block at current search entry through a simple strategy. As long as the candidate is selected, the related Dihedral transformation is performed on the domain block and the similarity measure can be calculated, which is regarded as the best one from the eight measures. Very little overhead is imposed. Therefore the encoding speed is almost the same as the method with no Dihedral transformations. Moreover, because most of the orientations can be correctly allocated, the quality of the reconstructed image is very close to the full search method.

Experimental results show that, the encoding speed of the proposed method is about 7 times faster than that of the full search method. The image quality (PSNR) of the full search method of Lena is 28.93 dB. For the proposed method, the quality is 28.75 dB and superior to that of the method without Dihedral transformations, which is 28.05 dB.

## 2. FRACTAL IMAGE COMPRESSION

The fundamental idea of fractal image compression is the Iteration Function System (IFS) in which the governing theorems are the Contractive Mapping Fixed-Point Theorem and the Collage Theorem [6]. Such IFS can hardly exist for a natural image because most of the sub-images are not directly similar to the whole image. To solve this problem, the idea of local self-similarity is adopted to form the Partitioned Iterated Function System (PIFS) [6].

Let  $f$  be a gray level image of size  $N \times N$ . Let the range pool be the set of the  $(N/L)^2$  non-overlapping blocks of size  $L \times L$  which is the coding unit. Let the contractivity of the fractal coding be a fixed quantity of 2. Thus, the domain pool makes up the set of  $(N - 16 + 1)^2$  overlapping blocks of size  $2L \times 2L$ . For each range block  $v$ , one searches in the domain pool to find the best matched domain block. At each search entry of the domain block  $u$ , it is first down-sampled to have the same size as that of the range block. For simplicity, let the down-sampled block be denoted by  $u$ . Also let the terms domain block and domain pool are referred to those down-sampled blocks. The full search method transforms  $u$  using the eight transformations in the Dihedral transformation on the pixel positions to increase the size of the domain pool. If the origin of the coordinate of  $u$  is

assumed to locate at the center of the block, the eight transformations  $T_k: k = 0, \dots, 7$  can be represented by the following matrices:

$$\begin{aligned} T_0 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, T_1 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, T_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, T_3 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \\ T_4 &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, T_5 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, T_6 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, T_7 = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}. \end{aligned} \quad (1)$$

Fractal coding also allows a contrast scaling  $p$  and a brightness offset  $q$  on the transformed domain blocks. Thus the fractal affine transformation  $\Phi$  of  $u(x, y)$  in  $D$  can be expressed as:

$$\Phi \begin{bmatrix} x \\ y \\ u(x, y) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & p \end{bmatrix} \begin{bmatrix} x \\ y \\ u(x, y) \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ q \end{bmatrix} \quad (2)$$

where the  $2 \times 2$  sub-matrix  $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$  is one of the dihedral transformations in Eq. (1).

Let the eight transformed blocks be denoted as  $u_k, k = 0, 1, \dots, 7$ , where  $u_0 = u$ . At each search entry, eight separated MSE computations are required to find the index  $d$  such that:

$$d = \arg \min \{ \text{MSE}((p_k u_k + q_k), v): k = 0, 1, \dots, 7 \} \quad (3)$$

where

$$\text{MSE}(u, v) = \frac{1}{L^2} \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} (u(i, j) - v(i, j))^2.$$

The quantities  $p_k$  and  $q_k$  can be computed directly by calculus optimization as:

$$p_k = \frac{[L^2 \langle u_k, v \rangle - \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} u_k(i, j) \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} v(i, j)]}{[L^2 \langle u_k, u_k \rangle - (\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} u_k(i, j))^2]} \quad (4)$$

$$q_k = \frac{1}{L^2} \left[ \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} v(i, j) - p_k \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} u_k(i, j) \right]. \quad (5)$$

As  $u$  runs over all of the domain blocks in  $D$ , the best match of  $v$  is found, the terms  $t_x$  and  $t_y$  in Eq. (2) are obtained. Together with  $d$  and the specific  $p$  and  $q$  corresponding to this  $d$ , the affine transformation Eq. (2) is constructed for the given range block  $v$ .

### 3. DIRECT ALLOCATING METHOD

For a given range block  $v$ , we search in the domain pool to find the best match. At each search entry  $u$ , eight similarity measures are required. Among these eight orientations of  $u$ , the most similar one and the corresponding index  $d$  from Eq. (3) should be recorded.

To speedup the computations, we propose a direct allocating method to predict the best candidate so that only one transformation and one MSE computation are required. The prediction is performed according to the three lowest AC coefficients of the DCT of  $u$  and  $v$ . Such AC's reflect essential edge properties of blocks and thus can be incorporated for similarity measures. The overhead is minor since only three lowest DCT coefficients are required to compute and the prediction is accomplished through table lookup mechanism. Therefore, approximate eight times speedup can be achieved. Also, since the three AC's coefficients possess most of the energy of a block, good prediction can be achieved so as to preserve reconstructed image quality.

At each search entry  $u$ , let  $u_k = T_k(u)$  for  $0 \leq k \leq 7$ , where  $T_k$  is given in Eq. (1). It is well known that  $T_1$ ,  $T_2$  and  $T_4$  are the generators of the group of Dihedral transformations in which every element can be constructed from these three elements through function compositions, *i.e.*,  $T_3 = T_1 \circ T_2$  and  $T_j = T_4 \circ T_{j-4}$  for  $j = 4, 5, 6, 7$ . As a consequence, it suffices to investigate the relation between the DCT coefficients of  $u$  and  $u_k$ ,  $k = 1, 2, 4$ . Intuitively, the transformation  $T_1$  can be regarded as the flip of  $u$  about the central vertical line. Let the origin of image coordinate be located at the upper left pixel and the horizontal and vertical axis be denoted as  $i$  and  $j$ , respectively. Let  $u$  be of size  $L \times L$ . By direct computation, the image block  $u_1$ ,  $u_2$  and  $u_4$  can be expressed explicitly in terms of  $u$  as:

$$u_1(i, j) = u(L - 1 - i, j) \quad (6)$$

where  $0 \leq i, j < L$ .  $T_2$  and  $T_4$  are the flips of  $u$  about the central horizontal line and the diagonal line, respectively. The formulas are given by:

$$u_2(i, j) = u(i, L - 1 - j), u_4(i, j) = u(j, i). \quad (7)$$

The DCT of  $u$ , denoted by  $U$ , is computed from the formula:

$$U(m, n) = \frac{2}{L} C_m C_n \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} u(i, j) \cos\left(\frac{(2i+1)m\pi}{2L}\right) \cos\left(\frac{(2j+1)n\pi}{2L}\right) \quad (8)$$

where  $m, n = 0, 1, \dots, L - 1$ , and

$$C_l = \begin{cases} 1/\sqrt{2}, & \text{if } l = 0 \\ 1, & \text{else} \end{cases}.$$

Let  $U_k$  be the DCT of  $u_k$ . For the case of  $L = 8$  and  $k = 0$ , the three lowest AC coefficients can be expressed by:

$$U(1, 0) = \frac{\sqrt{2}}{8} \sum_{j=0}^7 \sum_{i=0}^7 u(i, j) \cos \theta_i \tag{9}$$

$$U(0, 1) = \frac{\sqrt{2}}{8} \sum_{j=0}^7 \sum_{i=0}^7 u(i, j) \cos \theta_j \tag{10}$$

$$U(1, 1) = \frac{1}{4} \sum_{j=0}^7 \sum_{i=0}^7 u(i, j) \cos \theta_i \cos \theta_j \tag{11}$$

where  $\theta_i = (2i + 1)\pi/16$  and  $\theta_j = (2j + 1)\pi/16$ . Since  $\cos \theta_i$  is independent of  $j$ , the quantity  $U(1, 0)$  reflects the intensity energy variation between the left half and right half of image block  $u$ . Similar, the quantity  $U(0, 1)$  measures the intensity energy variation between the upper half and lower half of  $u$ . For  $u$  being partitioned into four equal sub-blocks,  $U(1, 1)$  measures the intensity energy variation between the two diagonal sub-blocks and the two sub-diagonal sub-blocks.

The Dihedral transformations affect the signs of the three coefficients. For  $T_1$ , the relation between  $U_1(1, 0)$  and  $U(1, 0)$  can be derived from the DCT formula (8) by:

$$U_1(1, 0) = \frac{\sqrt{2}}{L} \sum_{j=0}^{L-1} \sum_{i=0}^{L-1} u_1(i, j) \cos \theta_i = -\frac{\sqrt{2}}{L} \sum_{j=0}^{L-1} \sum_{i=0}^{L-1} u(i, j) \cos \theta_i = -U(1, 0).$$

The relation between  $U_1(0, 1)$  and  $U(0, 1)$  can also be derived as:

$$U_1(0, 1) = \frac{\sqrt{2}}{L} \sum_{j=0}^{L-1} \sum_{i=0}^{L-1} u_1(i, j) \cos \theta_j = \frac{\sqrt{2}}{L} \sum_{j=0}^{L-1} \sum_{i=0}^{L-1} u(i, j) \cos \theta_j = U(0, 1).$$

Similar, the relation between  $U_1(1, 1)$  and  $U(1, 1)$  is derived as:

$$U_1(1, 1) = \frac{\sqrt{2}}{L} \sum_{j=0}^{L-1} \sum_{i=0}^{L-1} u_1(i, j) \cos \theta_i \cos \theta_j = -\frac{\sqrt{2}}{L} \sum_{j=0}^{L-1} \sum_{i=0}^{L-1} u(i, j) \cos \theta_i \cos \theta_j = -U(1, 1).$$

As a conclusion, for the transformation  $T_1$  performed on  $u$ , both of the signs of  $U(1, 0)$  and  $U(1, 1)$  are changed while  $U(0, 1)$  remain unchanged. In a similar fashion, the signs of the coefficients  $U(0, 1)$  and  $U(1, 1)$  are changed for  $T_2$  performed on  $u$ . As a combination of  $T_1$  and  $T_2$ ,  $T_3$  changes the signs of  $U(1, 0)$  and  $U(0, 1)$ . Finally, a direct observation from (9) and (10) yields the effect of  $T_4$  performed on  $u$  exchanges the coefficients  $U(1, 0)$  and  $U(0, 1)$ . The relation of sign changes versus to the three Dihedral transformations are summarized in Table 1.

**Table 1. The sign changes of  $U$  when Dihedral transformations are performed on  $u$ .**

	$T_0$	$T_1$	$T_2$	$T_3$
$U(1, 0)$	+	-	+	-
$U(0, 1)$	+	+	-	-
$U(1, 1)$	+	-	-	+

The key idea of the proposed method is to make use the sign changes of the DCT coefficients as well as the exchange of  $U(1, 0)$  and  $U(0, 1)$  versus to the Dihedral transformations. First, we change the signs of the lowest three AC coefficients of  $u$  according to Table 1 such that signs of them are all the same or opposite as those of  $v$ . Second, we exchange  $U(1, 0)$  and  $U(0, 1)$  if necessary such that the order of the magnitudes of  $U(1, 0)$  and  $U(0, 1)$  coincides to that of  $V(1, 0)$  and  $V(0, 1)$ , *i.e.*,

$$\begin{aligned} |U(1, 0)| \geq |U(0, 1)| & \text{ if } |V(1, 0)| \geq |V(0, 1)| \\ |U(1, 0)| \leq |U(0, 1)| & \text{ if } |V(1, 0)| < |V(0, 1)|. \end{aligned} \quad (12)$$

As the appropriate transformation  $T$  is selected, both the three lowest AC coefficients of the DCT of  $Tu$  and  $v$  will have the same sign relation and magnitude order. Thus, it indicates that  $Tu$  and  $v$  will have the similar edge properties in the sense that the left-right, up-down and diagonal-subdiagonal intensities variations are similar. For fractal coding, the similarity measure, up to certain contrast scaling and brightness offset, between these two blocks will be small.

Let  $b$  be a block of size  $L \times L$  and the DCT of  $b$  be denoted by  $B$ . The sign pattern  $\hat{b}$  of  $b$  is defined as:

$$\hat{b} = (\hat{b}_1, \hat{b}_2, \hat{b}_3) = (\text{sgn}(B(1, 0)), \text{sgn}(B(0, 1)), \text{sgn}(B(1, 1)))$$

where

$$\text{sgn}(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ -1, & \text{else} \end{cases}.$$

The sign difference of the two blocks  $u$  and  $v$  is defined as:

$$\rho(u, v) = (\rho_1(u, v), \rho_2(u, v), \rho_3(u, v))$$

where

$$\rho_j(u, v) = \begin{cases} 1, & \text{if } \hat{u}_j \neq \hat{v}_j \\ 0, & \text{else} \end{cases}, \quad j = 1, 2, 3.$$

It is clear,  $\rho(u, v)$  indicates the positions in which the lowest three AC coefficients of  $u$  and  $v$  have different signs. The direct allocation of the Dihedral transformation is performed based on this quantity, which is discussed in the following four cases.

**Case 1:**  $\rho(u, v) = (0, 0, 0)$ ,  $\rho(u, v) = (1, 1, 1)$ .

For  $\rho(u, v) = (0, 0, 0)$ , all the three coefficients of  $u$  and  $v$  have the same signs. If the magnitude order coincides as stated in Eq. (12), then  $u$  and  $v$  tend to have similar edge properties up to certain contrast scaling  $p$  and brightness offset  $q$ . Therefore, this  $u$ , without Dihedral transformation, is a good candidate for the best match of  $v$  and only  $T_0$  is required for  $u$ . On the other hand, if the magnitude order does not coincide, we need to exchange  $U(1, 0)$  and  $U(0, 1)$  such that the resulting order satisfies Eq. (12). Thus,  $T_4$  is the preferred candidate. For  $\rho(u, v) = (1, 1, 1)$ , all of the three coefficients have opposite

signs. It is known DCT is linear in scalar multiplication, so we have  $\rho(-u, v) = (0, 0, 0)$ . Since the minus sign of  $-u$  will be absorbed as part of the contrast scaling  $p$ , this  $u$  can be regarded to have same edge properties as the one in the above case of  $\rho(u, v) = (0, 0, 0)$ . Consequently,  $T_0$  is selected if the magnitude order coincides, or  $T_4$  is selected otherwise.

**Case 2:**  $\rho(u, v) = (1, 0, 1)$ ,  $\rho(u, v) = (0, 1, 0)$ .

$\rho(u, v) = (1, 0, 1)$  indicates that the pair  $U(1, 0)$  and  $V(1, 0)$  as well as the pair  $U(1, 1)$  and  $V(1, 1)$  have opposite signs. According to Table 1, if we perform  $T_1$  on  $u$ , the signs of both  $U(1, 0)$  and  $U(1, 1)$  will be changed, *i.e.*,  $\rho(T_1u, v) = (0, 0, 0)$ . If the magnitude order coincides,  $T_1$  is the transformation selected, otherwise, we need an additional  $T_4$  to exchange  $U(1, 0)$  and  $U(0, 1)$  to satisfy (12). Therefore  $T_5 = T_4 \circ T_1$  is selected. If  $\rho(u, v) = (0, 1, 0)$ , we have  $\rho(-u, v) = (1, 0, 1)$  and  $\rho(-T_1u, v) = (0, 0, 0)$ . Again, since the negative sign will be absorbed in  $p$ , similar argument addressed in Case 1 asserts  $T_1$  or  $T_5$  is allocated according to the magnitude order.

**Case 3:**  $\rho(u, v) = (0, 1, 1)$ ,  $\rho(u, v) = (1, 0, 0)$ .

$\rho(u, v) = (0, 1, 1)$  indicates that the pair  $U(0, 1)$  and  $V(0, 1)$  as well as the pair  $U(1, 1)$  and  $V(1, 1)$  have opposite signs. According to Table 1,  $T_2$  is the selected transform. Therefore,  $T_2$  or  $T_6 = T_4 \circ T_2$  is preferred according to the magnitude order. Also, either  $T_2$  or  $T_6$  is selected for the case of  $\rho(u, v) = (1, 0, 0)$ .

**Case 4:**  $\rho(u, v) = (1, 1, 0)$ ,  $\rho(u, v) = (0, 0, 1)$ .

Similar,  $T_3$  will change the signs of  $U(1, 0)$  and  $U(0, 1)$ . Therefore,  $T_3$  or  $T_7 = T_4 \circ T_3$  is selected for both  $\rho(u, v) = (1, 1, 0)$  and  $\rho(u, v) = (0, 0, 1)$  according to the magnitude order.

As derived in the four cases above, we can directly allocate the Dihedral transformation which is a good prediction of the best one out of the eight orientations. Therefore, the eight computations of similarity measure at each search entry are reduced to one. Theoretically, there will be a speedup ratio of eight.

#### 4. EXPERIMENTAL RESULTS

The proposed method is implemented using Borland C++ Builder 6.0 running on PC with AMD Athlon™ 1.4 GHz CPU. In the simulation, 4 images, Lena, Pepper, F16 and Baboon, are tested. The statistics of various image sizes of  $64 \times 64$ ,  $128 \times 128$  and  $256 \times 256$  and coding units of  $4 \times 4$ ,  $8 \times 8$  and  $16 \times 16$  are analyzed. Three methods, the full search, the method without Dihedral transformation and the proposed methods, are all tested and compared first. Let the later two be referred to as No-Dihedral and One-Dihedral methods, respectively. The reports are focused on the speedup ratio and reconstructed image quality among the three methods. Also, the prediction efficiency of the proposed method is demonstrated. The term speedup ratio is the ratio of the encoding time of the full search method over that of the proposed method. The reconstructed image quality is measured by Peak Signal to Noise Ratio (PSNR).

The encoding times of the three methods are given in Table 2. For Lena of size  $256 \times 256$  with coding unit  $8 \times 8$ , the speedup ratio for No-Dihedral and One-Dihedral methods are 6.77 and 6.68, respectively. As observed, the speedup ratios of the two methods are almost the same for all the other test images of various image sizes with various coding units.

**Table 2. The encoding time and the speedup ratio for Lena for three methods.**

Image size	Coding unit	Full search encoding time	No-Dihedral		One-Dihedral	
			Encoding time	Speedup Ratio	Encoding time	Speedup Ratio
64	4	14.040	2.204	6.370	2.223	6.316
	8	9.504	1.412	6.731	1.422	6.684
	16	4.236	0.611	6.933	0.611	6.933
128	4	253.795	40.268	6.303	39.858	6.367
	8	202.711	29.933	6.772	29.853	6.790
	16	153.742	21.080	7.293	21.071	7.296
256	4	4259.695	674.130	6.319	679.497	6.269
	8	3679.511	550.862	6.680	543.512	6.770
	16	3136.886	454.203	6.906	455.225	6.891

At each search entry, the computations required to find the best match among the eight orientations consists of the operations of down-sampling, Dihedral transformation, contrast and brightness adjustment, and MSE computations. For No-Dihedral method, not all the operations are reduced eight times, *e.g.*, the sub-sampling and contrast and brightness adjustment. This is why eight times speedup ratio can not be achieved. The One-Dihedral method can have the same speedup ratio as that of the No-Dihedral method. This is because the DCT-based direct allocating method uses only the lowest three DCT coefficients together with a simple table lookup mechanism which produce very low overhead.

Fig. 1 shows the reconstructed image qualities of  $256 \times 256$  Lena. Clearly, the One-Dihedral method has only 0.180 dB decay while the No-Dihedral method has 0.884 dB decay. The image quality of the proposed method is approximately the same as that of the full search method for all the test images. Examples of the reconstructed images for  $256 \times 256$  Lena are shown in Fig. 2. As observed, the One-Dihedral method does provide better image quality in comparison to the No-Dihedral method.

The reason that the One-Dihedral method preserves image quality is that the lowest three AC coefficients contain a great portion of energy of a block. It reflects global edge properties and therefore the performance of the prediction can be satisfied. To demonstrate this characteristic, we count the number of range blocks each of which finds the best matched domain block and correctly predicts the orientations as found in full search method. Such range blocks are called hit blocks. Table 3 shows the percentage of hit blocks of  $256 \times 256$  images with various coding units for both methods. For Lena with coding unit  $8 \times 8$ , the One-Dihedral method can correctly predict the best orientation up to 71%. In contrast, No-Dihedral method achieves only 21.39%. Such statistics reveals

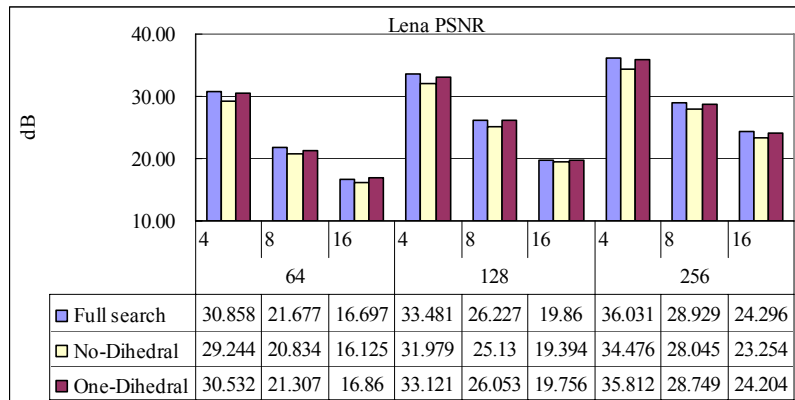
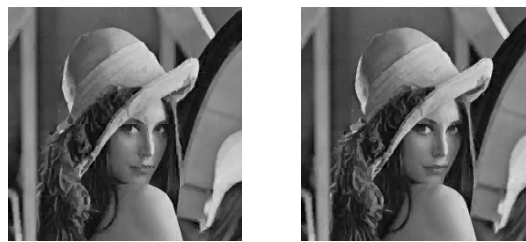


Fig. 1. The PSNR for the three methods of Lena.



(a) No-dihedral method. (b) One-dihedral method.

Fig. 2. Reconstructed images for  $256 \times 256$  Lena of coding unit  $8 \times 8$ .

**Table 3. The percentage of hit blocks of  $256 \times 256$  images for One-Dihedral and No-Dihedral method with various coding units.**

Coding unit \ Test Image	4		8		16	
	One-Dih	No-Dih	One-Dih	No-Dih	One-Dih	No-Dih
Lena	74.95%	18.04%	71.00%	21.39%	67.58%	26.95%
Pepper	78.08%	16.02%	75.59%	21.48%	66.02%	20.70%
F16	73.10%	16.06%	66.41%	22.46%	69.53%	22.66%
Baboon	62.01%	13.79%	49.80%	14.26%	47.27%	19.14%

that the eight orientations are indeed required to improve the image quality for fractal image compression.

For each of the non-hit blocks, although the best domain block is not found, the secondary-best block can still be found with the One-Dihedral method embedded in the searching strategy. Let the degree of similarity of a given range block be the MSE of itself and the matched domain block. The degree of a hit block certainly has the same MSE value as that in the full search method. For non-hit blocks, the degree deviates. Fig. 3 shows the percentage of  $8 \times 8$  range blocks for which the quantity  $\Delta\text{MSE}$ , defined as the MSE deviations of the proposed method from the full search method, does not exceed a certain values. For Lena, as indicated in Fig. 3, 93.75% of the range blocks find their

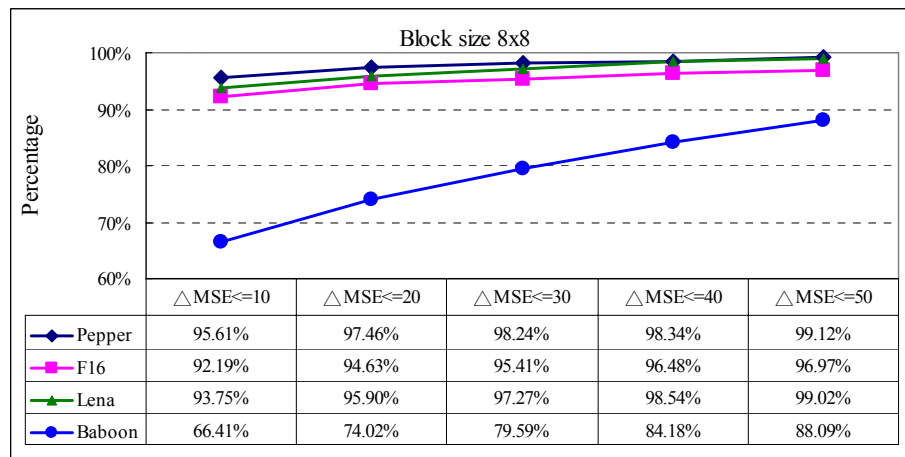


Fig. 3. Percentage of range blocks with MSE deviations using  $8 \times 8$  coding units.

**Table 4. Coding efficiencies of various methods of  $256 \times 256$  Lena with coding unit  $8 \times 8$ .**

Encoding method	Speedup ratio	PSNR difference
Proposed One-Dihedral method	6.77	- 0.18
Fast algorithm using DCT inner product [11]	3.47	+ 0.03
Algorithm using DCT inner product with Zonal filter type 4 [11]	5.90	- 0.72
Fast algorithm using Hadamard transform [12]	2.73	0.0
No-Dihedral method	6.68	- 0.68

matches with  $\Delta \text{MSE}$  less than 10. As a consequence, the proposed One-Dihedral method can preserve image quality.

The coding efficiencies of the proposed methods and the other four methods using  $256 \times 256$  Lena with coding unit  $8 \times 8$  are listed in Table 4. The PSNR difference shown in the table is defined as the difference of PSNR between the full search method and the encoding method used. As shown in the first row, the proposed method has 6.77 speedup ratio with only 0.18dB quality decay. Although the fast algorithm using DCT inner product has the least PSNR difference, it has only 3.5 speedup ratio. On the other hand, if further zonal filter is added, the speedup ratio using DCT inner product reaches 5.9 but the image quality decay grows up to 0.72 dB. The fast algorithm using Hadamard transform has no quality decay, but it has only 2.7 speedup ratio. For the No-Dihedral method shown in the last row, although the speedup ratio is the same as that of the proposed method, the quality decay is 0.68dB. Therefore the proposed method does speed up the encoder and preserve the image quality as well.

## 5. CONCLUSION

Full search fractal image compression can obtain good image quality due to the exhaust search in a large codebook to find the best match. The penalty is that it is time con-

suming. No-Dihedral method can speedup the encoder but it also produces quality decay. In this paper, we propose a direct allocating method to predict the best Dihedral transformation based on the lowest three DCT coefficients. This One-Dihedral method invokes very low overhead and thus it has the same encoding speed as that of the No-Dihedral method. Moreover, since the prediction is superior, the image quality is almost the same as that of the full search method.

## REFERENCES

1. H. O. Peitgen, J. M. Henriques, and L. F. Penedo, *Fractals in the Fundamental and Applied Sciences*, Elsevier Science Publishing Company Inc., New York, 1991.
2. M. F. Barnsley and S. Demko, "Iterated function systems and the global construction of Fractals," in *Proceeding of Royal Society*, Vol. A399, 1985, pp. 243-275.
3. E. Jacquin, "Image coding based on a Fractal theory of iterated contractive image transformations," *IEEE Transactions on Signal Processing*, Vol. 1, 1992, pp. 18-30.
4. E. W. Jacobs, Y. Fisher, and R. D. Boss, "Image compression: A study of the iterated transform method," *IEEE Transactions on Signal Processing*, Vol. 29, 1992, pp. 251-263.
5. M. F. Barnsley, "Methods and apparatus for image compression by iterated function systems," United States Patent No. 4941193, 1990.
6. Y. Fisher, *Fractal Image Compression, Theory and Application*, Springer-Verlag, New York, 1994.
7. B. Wohlberg and G. de. Jager, "A review of the fractal image coding literature," *IEEE Transactions on Image Processing*, Vol. 8, 1999, pp. 1716-1729.
8. H. Hartenstein and D. Saupe, "Lossless acceleration of fractal image encoding via the fast Fourier transform," *Signal Processing: Image Communication*, Vol. 16, 2000, pp. 383-394.
9. J. Cardinal, "Fast fractal compression of greyscale images," *IEEE Transactions on Image Processing*, Vol. 10, 2001, pp. 159-164.
10. S. Furaoa and O. Hasegawa, "A fast no search fractal image coding method," *Signal Processing: Image Communication*, Vol. 19, 2004, pp. 393-404.
11. J. H. Jeng, T. K. Truong, and J. R. Sheu, "Fast fractal image compression using the Hadamard transform," in *IEE Proceedings of Vision, Image Signal Processing*, Vol. 147, 2000, pp. 571-574.
12. T. K. Truong, J. H. Jeng, I. S. Reed, P. C. Li, and A. Q. Li, "A fast encoding algorithm for fractal image compression using the DCT inner product," *IEEE Transactions on Image Processing*, Vol. 9, 2000, pp. 529-535.

**Duh-Der Jyh (杜德智)** received the B.S. degree in Electrophysics in 1981 from National Chiao Tung University, the M.S. degrees in Electrical Engineering in 1983 from Tatung University and the Ph.D. degree in Computer Engineering from Yuan Ze University in 2005. From 1983. He has been working as a senior system engineering at the Information and Communication Research Division of Chung Shan Institute of Science and Technology (CSIST). His research interests include image processing, pattern recognition, image compression and content retrieval.

**Jyh-Horng Jeng (鄭志宏)** received the B.S. and M.S. degree in mathematics from Tamkang University, Taiwan in 1982 and 1984, respectively, and the Ph.D. degree in mathematics from State University of New York at Buffalo in 1996. He was as Senior Research Engineer with Chung Shan Institute of Science and Technology (CSIST) Taiwan, from 1984 to 1992. Currently, he is a Professor in the Department of Information Engineering, I-Shou University, Taiwan. His research interests include Multimedia Application, Content-based Retrieval, Data Mining, Soft Computing, Machine Learning, Information Security and VLSI architecture Design.

**Shu-Yuan Chen (陳淑媛)** received the B.S. degree in Electrophysics in 1980, the M.S. and Ph.D. degrees both in Computer Engineering in 1982 and 1990, all from National Chiao Tung University. From 1994, she has been an Associate Professor in the Department of Computer Science and Engineering at Yuan Ze University. From 2002, she has been a professor at the same department. Her research interests include bioinformatics, image processing, pattern recognition, and image content retrieval.