

Quality Adapted Fractal Image Compression for Multimedia Communication

D. J. Duh

Dept. of Computer Science and
Information Engineering,
Ching-Yun University, Taoyuan
County 320, Taiwan
+886-3-250-3013 ~ 7701
duby@saturn.yzu.edu.tw

J. H. Jeng

Dept. of Information Engineering,
I-Shou University,
Kaohsiung County 840, Taiwan
+886-7-657-7711 ~ 6523
jjeng@isu.edu.tw

S. Y. Chen

Dept. of Computer Science and
Engineering, Yuan-Ze University,
Taoyuan County 320, Taiwan
+886-3-463-8800 ~ 2357
cschen@saturn.yzu.edu.tw

ABSTRACT

In multimedia communication, the communication bandwidth is a key factor for real time data transfer. Image compression, such as fractal compression, is used for image communication. Encoding speed is a key factor rendering the fractal image compression unsuitable for real time multimedia communication applications. In this paper, we propose a mechanism for speed-quality control, which is capable of speeding the encoder by a specified factor with reasonable quality decay. This control strategy enables us to control the compression ratio and the reconstructed image quality in advance depending on the need of multimedia communication. The speedup method is accomplished through a classification scheme in the frequency domain using only 2 DCT coefficients, which produces very little overhead. Image qualities can be preserved as well since the scheme makes use of edge property of image blocks, in which the fractal similarity is embedded. Simulations using number of classes ranging from 1 to 30 are performed. When the number of classes is within 30, the speedup ratio can be approximately achieved with very little quality decay. As the number of classes is set to 30, the actual encoding speed can reach up to 29.1 times and there is only 0.57 dB quality decay.

Categories and Subject Descriptors

I.4.2 [IMAGE PROCESSING AND COMPUTER VISION]:
Compression (Coding)-Approximate methods.

General Terms

Algorithms

Keywords: Fractal Image Compression, Speed-Quality Control, Classification, DCT, MSSIM.

1. INTRODUCTION

In multimedia communication, due to the huge amount of data and limited bandwidth, real-time image transfer is still an ongoing study. Image compression is needed for multimedia image communication. 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 [1,2]. The fractal transform for image compression was introduced by M. F. Barnsley and S. Demko [3].

The first practical fractal image compression scheme, which utilized block-based transformations and an exhaustive search strategy, was introduced by A. E. Jacquin [4], E. W. Jacobs, Y. Fisher and R. D. Boss [5]. Their approach was an improved version of the system patented by Barnsley [6,7]. One of the main drawbacks of fractal coder is the computational complexity, i.e., the encoding speed. Commonly used methods to speed up the encoder are related to the partition or classification of image blocks [8-12] or limiting the search space [13,14].

Nappi *et al.* [11] concluded that there are two major categories for speedup techniques: classification based techniques and feature vector based techniques. In Y. Fisher's classification method [9], a given image block was divided into four quadrants. For each quadrant, its average and variance were computed. According to certain combination of these values, 72 classes were constructed. This method reduced the searching space efficiently. However, it required large amount of computations and the arrangement of these 72 classes was complicated.

In Z. Wang, D. Zhang and Y. Yu [12], four types of range blocks were defined based on the edge of the decoded image. They used a hybrid type of coding mechanism to achieve higher compression ratios while maintaining reasonable image quality. However, they did not provide any guideline of threshold determination, the speedup ratio varied unstably from about 1.6 to 5. In C. K. Lee *et al.* [13], the search of domain blocks was limited according to the variance of the block. Although they provided a 1 to 10 times of speedup ratio, the ratio was image dependent. In C. M. Lai *et al.* [14], a single kick-out condition was proposed to speed up the encoder. They needed to combine the DCT inner product algorithm proposed in [15] to retain a speedup ratio of 4.

In this paper, we propose an edge-based classification scheme to speed up the fractal encoder with speed-quality control. A range of speedup ratios can be achieved with fair quality decay. The classifier utilizes only 2 DCT coefficients, the lowest horizontal and the vertical coefficients, to define an edge value. These two coefficients represent the strength of the horizontal and vertical energy variations in the block. With a proper criterion for threshold, image blocks are classified into classes of equal number of members according to their edge indices. In the encoding process, only the blocks of the same class are required to perform the similarity measurement.

Since the searching space is reduced, the encoding speed can be improved. Also, since only 2 DCT coefficients are involved, a very small amount of overhead is imposed, in comparison with

the complexity of the encoder. In the meanwhile, the structure of the classifier is simple in the sense of fractal coder because it merges the Dihedral transformations together. The thresholds for the classifier are adaptively determined so that the speedup ratio is proportional to the number of classes and independent to the given image. Although it is obvious that the quality decay of the retrieved images grows as the number of class increases, the quality can still be retained in a reasonable range. This is because the classifier is designed according to the edge property of image blocks. As a consequence, a mechanism of speed-quality control can be achieved.

2. FULL SEARCH FRACTAL IMAGE COMPRESSION

The fundamental idea of fractal image compression is based on the Iteration Function System (IFS) with two governing theorems which are the Contractive Mapping Fixed-Point Theorem and the Collage Theorem [9].

Fractal image compression is an inverse problem, i.e., for the given set S , find the IFS which has S as its attractor. It should be noted that, when S is a natural image, such IFS can hardly exist 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) in which the contractive maps w_i is defined only on D_i where $D_i \subset X$ for $i=1, \dots, n$.

Let f be a gray level image of size $N \times N$. Let the range pool R be the set of the $(N/8)^2$ non-overlapping blocks of size $L \times L$ which is the size of encoding 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 the case of a 256×256 image with 8×8 encoding size, the domain pool contains 58081 blocks of size 16×16 and the range pool contains 1024 blocks of size 8×8 . For each range block v in R , one searches in the domain pool D to find the best match, i.e., the most similar domain block. The parameters describing this fractal affine transformation form the fractal compression codes of v .

At each search entry, the domain block is first down-sampled to 8×8 and denoted as u . The down-sampled block is transformed subject to the eight transformations in the Dihedral group on the pixel positions. If the origin 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.

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

The eight transformed blocks are denoted as u_k , $k = 0, 1, \dots, 7$, where $u_0 = u$. The transformations T_1 and T_2 correspond to the flips of u along the vertical and horizontal lines, respectively. T_3 is the flip along both the vertical and horizontal lines. T_4 , T_5 , T_6 and T_7 are the

transformations of T_0 , T_1 , T_2 and T_3 , respectively, performed by an additional flip along the main diagonal line.

In the idea of fractal coding, a contrast scaling and a brightness offset are allowed to operate on the transformed domain blocks. Thus the fractal affine transformation Φ 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 2×2 sub-matrix $\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ is one of the dihedral

transformations in equation (1).

In each search entry, eight separated computations of Mean Square Error (MSE) 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

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

Here p_k and q_k can be computed directly by calculus optimization method 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 58081 domain blocks in D to find the best match, the terms t_x and t_y in equation (2) can be obtained. Together with d and the specific p and q corresponding to this d , the affine transformation in equation (2) is found for the given range block v . In practice, t_x , t_y , p , q and d can be encoded using 8, 8, 5, 7 and 3 bits, respectively, which are regarded as the compression code of v . Finally, as v runs over all of the 1024 range blocks in R , the encoding process is completed.

To decode the image, we first make up the 1024 affine transformations from the compression codes and choose an arbitrary image as the initial image. Then, we perform the 1024 affine transforms on the image to obtain a new image, and proceed recursively. According to Partitioned Iteration Function Theorem (PIFS), the sequence of images will converge. The stop criterion of the recursion is designed according to user's application. The final image is the decoded image of fractal coding.

3. CLASSIFICATION SCHEME BASED ON EDGE PROPERTY

The proposed local search strategy is designed through a classification scheme based on the edge property of blocks. Such a strategy can speed up the encoder and preserve image quality. The classifier operates in the frequency domain and makes use of edge property incorporated with the Dihedral transformations.

Let u be a given image block of size $L \times L$, The DCT of u , denoted by U , is computed from the formula:

$$U(m,n) = \frac{2}{L} C_m C_n \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} u(i,j) \cos\left(\frac{(2i+1)m\pi}{2L}\right) \cos\left(\frac{(2j+1)n\pi}{2L}\right) \quad (6)$$

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

$$C_k = \begin{cases} 1/\sqrt{2}, & \text{if } k=0 \\ 1, & \text{else} \end{cases}$$

Taking $L=8$ as an example, we have 64 DCT coefficients. Among the 64 DCT coefficients, lower coefficients possess most of the energy of the block. Higher coefficients contain only finer shape information. If the lower coefficients of two blocks are alike, the similarity measure between them won't be large due to the energy concentration. On the other hand, if two blocks are similar except for some content details, the magnitudes of their lower DCT coefficients will be roughly the same. Among the lower coefficients, the quantity $|U(0,0)|$ which contains the main brightness information of the block can be integrated into q_k in equation (5) during the calculation of fractal code. Therefore, the lower coefficients excluding the DC term are sufficient for the similarity measure of two blocks.

Let $V=U(1,0)$ and $H=U(0,1)$. For the case of $L=8$, we have:

$$V = \frac{\sqrt{2}}{8} \sum_{j=0}^7 \sum_{i=0}^7 u(i,j) \cos \theta_i \quad (7)$$

$$H = \frac{\sqrt{2}}{8} \sum_{j=0}^7 \sum_{i=0}^7 u(i,j) \cos \theta_j \quad (8)$$

where $\theta_i = (2i+1)\pi/16$ and $\theta_j = (2j+1)\pi/16$ with $i, j = 0, 1, \dots, 7$. It is clear that $\cos \theta_i > 0$ for $i = 0, 1, 2, 3$ and $\cos \theta_i < 0$ for $i = 4, 5, 6, 7$. Also, since $\cos \theta_i$ is independent of j , the quantity V reflects the energy variation between the left half and the right half of the block u . Similarly, the quantity H measures the energy variation between the upper half and the lower half of u .

The function $\cos \theta_i$ is an odd function with respect to the middle vertical line of $u(i,j)$. Thus the value of V is negated if u is transformed by T_1 given in equation (1). Since H is independent of $\cos \theta_i$, it will remain unchanged. Similarly, if the horizontal flip T_2 is applied, V is unchanged and H is negated. The transformation T_3 flips u both vertically and horizontally, therefore both V and H are negated. The additional transformation T_4 for which $T_{k+4} = T_4 T_k$, $k = 0, 1, 2, 3$, flips a block about the main diagonal line, so it will exchange the indices i and j for $u(i,j)$. According to equation (6), the quantities H and V are interchanged. The detailed changes of V and H corresponding to the eight Dihedral transformations are shown in Table.1.

Table 1. The changes of signs and interchange relation of V and H after applying the Dihedral transformations.

	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7
V	+V	-V	+V	-V	+H	-H	+H	-H
H	+H	+H	-H	-H	+V	+V	-V	-V

The design of the classifier is based on the relations between the magnitudes of H and V of given blocks, in which the eight Dihedral transformations are incorporated. First, we calculate

H and V of a block. Then we perform the appropriate transformations according to Table 1 such that the new values of H and V of the transformed block are both positive and satisfy $|V| \leq |H|$. That is, there are projected to the 45° fan area from the H -axis in the first quadrant of $H-V$ coordinate system. With this incorporation of the Dihedral transformations, we design the classifier according to the quantity $|V|/|H|$ which will serve as an indication of edge property of an image block.

Now we will partition the 45° fan area into C sub-fan areas according to the argument above. Let E_u be the edge value defined as:

$$E_u = \min\left\{\frac{|V|}{|H|}, \frac{|H|}{|V|}\right\} \quad (9)$$

Let $\tau = \{\tau_0, \tau_1, \dots, \tau_C\}$ be the set of $C+1$ predefined thresholds. The classification scheme will assign a block u to one of the C classes according to the magnitude E_u and the thresholds τ by the criterion:

$$c_u = j \quad \text{if } \tau_j \leq E_u < \tau_{j+1}, \quad (10)$$

for some $j \in \{0, 1, \dots, C-1\}$

If E_u is small, the block u tends to have strong vertical or horizontal edges. On the other hand, if E_u is close to 1, u tends to have strong diagonal or sub-diagonal edges. The block diagram of the classifier is depicted in Fig. 1, and the steps are summarized as follows:

Compute V and H of u from equation (7) and equation (8).

Compute the edge value E_u from equation (9).

Search for τ_j such that $\tau_j \leq E_u < \tau_{j+1}$ and assign the index of the block as j .

It is obvious that the classification scheme is simple and only few computations are required to calculate the lower DCT coefficients. The overhead of this classifier is very small compared to the fractal coding process. Moreover, since the classifier is designed according to the edge property, the image quality can be preserved.

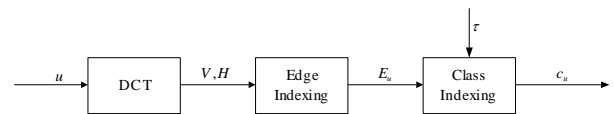


Figure 1. Block diagram of the classification scheme based on edge property.

4. SPEED-QUALITY CONTROL

The classification scheme based on edge property given in Section 3 is now applied to fractal image encoder to accomplish speed-quality control. Given a desired speedup ratio, the encoder will attain the ratio with reasonable quality decay. For a given ratio C , we set the number of classes as C and calculate the thresholds $\tau = \{\tau_0, \tau_1, \dots, \tau_C\}$ from the domain pool D with $\tau_0 = 0$ and $\tau_C = 1$. The thresholds are adaptively determined so as to partition D into C sub-pools of nearly equal number of members, which are denoted as D_j , $0 \leq j < C$. Each domain block is assigned a class index respectively. To encode a range block, we first determine its class index according to the same thresholds τ and find the best match from those domain blocks with the same class index. The block

diagram of fractal image encoder with speed-quality control is given in Fig. 2, where the Threshold and Domain Sub-pool Determination block is the pre-processing stage.

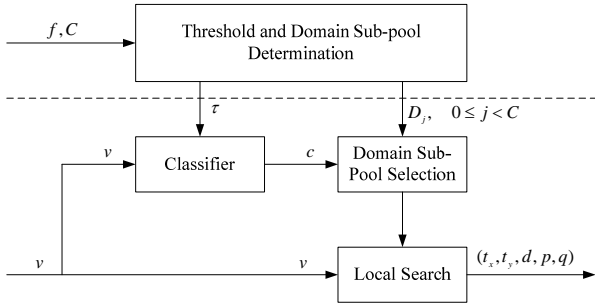


Figure 2. Block diagram of fractal image encoder using the proposed scheme to achieve speed-quality control.

The number of MSE computations for finding the best match, which is proportional to the speedup ratio, can be estimated. Let M_D and N_R denote the numbers of blocks in the domain pool D and range pool R , respectively. Also, let m_j and n_j denote the numbers of blocks after classification in each domain sub-pool and range sub-pool, respectively. The total number of MSE computations for the encoder can be estimated by the following equation:

$$\sum_{j=0}^{C-1} m_j \times n_j \approx \sum_{j=0}^{C-1} \frac{M_D}{C} \times n_j = \frac{1}{C} \times M_D \times \sum_{j=0}^{C-1} n_j = \frac{1}{C} M_D \times N_R$$

Since the encoding time essentially comes from the number of MSE computations and $M_D \times N_R$ is the total number of MSE computations for the full search method, we will achieve the speedup ratio of C theoretically.

For small C , since the overhead of the amount of MSE computations dominates and the domain pool is almost equally partitioned, the desired speedup ratio can be achieved. On the other hand, when C is large, we might not be able to equally partition the domain pool. Therefore, along with the overhead of the classifier and the pre-processing unit, the speedup ratio can no longer be guaranteed. It will end up to a saturation value.

5. EXPERIMENTAL RESULTS

The proposed speed-quality control mechanism for fractal coding is simulated and verified. The programs are implemented using Borland C++ Builder 5.0 running on Microsoft Windows 2000, Pentium III 1 GHz platform. In the simulation, 4 images, Lena, Pepper, F16 and Baboon, are tested and the statistics are analyzed.

The speedup ratio is defined as the ratio of the encoding time of the full search method to that of the proposed method. The distortion or error between the original image $f(i, j)$ and the decoded image $\hat{f}(i, j)$ caused by lossy compression process is measured in Peak Signal to Noise Ratio (PSNR) defined by:

$$\text{PSNR} = 10 \log_{10} \left(\frac{255^2}{\text{MSE}} \right)$$

The quantity C , i.e., the number of classes, is the key term in the simulation. The full search method is equivalent to the proposed method with $C=1$. Fig. 3 shows the speedup ratio for C ranging from 1 to 30. As illustrated, the speedup ratios nearly coincide with the numbers of classes for all of the four images. The quality of the decoded images versus the number of classes is shown in Fig. 4. It indicates that as C increases, only small amount of quality decay is imposed. The curve of speed-quality control is depicted in Fig. 5. It reveals that high speedup ratios

can be achieved by specifying the quantity C with only little quality decay. The detailed simulation results of Lena are given in Table 2 with $C=1$ corresponding to the full search method. As shown in the table, when C is chosen to be 30, the encoding speed is about 29 times faster with only 0.65 dB of quality decay in comparison with that of the full search method.

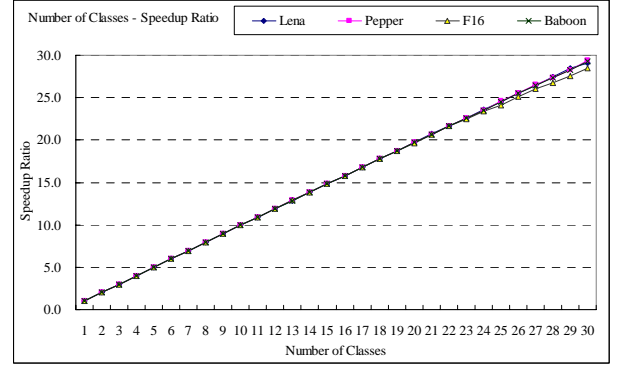


Figure 3. Speedup ratio for number of classes ranging from 1 to 30.

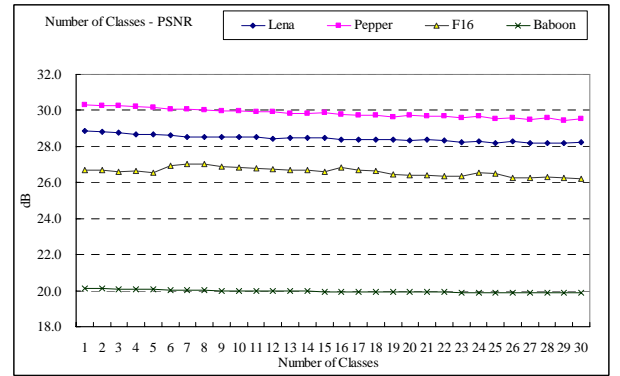


Figure 4. PSNR of the decoded images for number of classes ranging from 1 to 30.

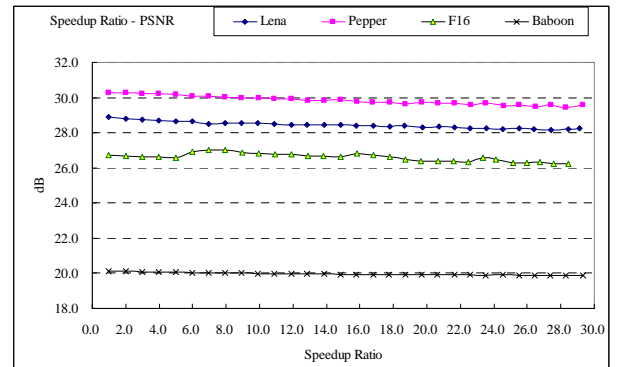


Figure 5. Speed-quality curve for number of classes ranging from 1 to 30.

Table 2. Encode time, Speedup ratio and PSNR of Lena for small C.

Image	Lena		
	Encode Time (sec)	Speedup Ratio	PSNR
1	1650.530	1.000	28.880
2	826.107	1.998	28.820
3	551.413	2.993	28.751
4	413.885	3.988	28.682
5	331.407	4.980	28.650
6	276.468	5.970	28.636
7	237.000	6.964	28.515
8	207.909	7.939	28.530
9	184.776	8.933	28.543
10	166.349	9.922	28.526
11	151.408	10.901	28.523
12	138.880	11.885	28.437
13	128.344	12.860	28.474
14	119.131	13.855	28.453
15	111.341	14.824	28.464
16	104.580	15.783	28.380
17	98.311	16.789	28.388
18	92.934	17.760	28.369
19	88.337	18.685	28.397
20	83.580	19.748	28.319
21	79.745	20.698	28.376
22	76.239	21.649	28.320
23	73.166	22.559	28.258
24	70.080	23.552	28.263
25	67.277	24.533	28.206
26	64.713	25.505	28.270
27	62.540	26.392	28.180
28	60.177	27.428	28.172
29	58.033	28.441	28.200
30	56.722	29.099	28.233

6. CONCLUSION

In this paper, the mechanism of speed-quality control of fractal image compression for multimedia communication is proposed. The control is designed through a classification scheme based on edge property of image blocks. The achievement of the control is owing to the following four main issues: 1) The classification scheme causes low overhead since only 2 DCT coefficients are considered. 2) The classification scheme is efficient since the fractal Dihedral transformations are incorporated. 3) Speedup ratios are guaranteed up to some extent because of adaptive threshold. 4) Image quality can be retained because edge property is adopted. Simulation results

show that the speedup ratios can be attained with fair quality decays by specifying the corresponding number of classes less than 30. The controllability fades for higher speedup ratios and is dependent on the content of the image itself. This control strategy enables us to control the compression ratio and the reconstructed image quality in advance depending on the need of multimedia communication.

7. REFERENCES

- [1] Peitgen, H.O., Henriques, J.M., and Penedo, L.F. 1991, Fractals in the fundamental and applied sciences, New York, Elsevier Science Publishing Company Inc.
- [2] Crilly, J., Earnshaw, R.A., and Jones, H. 1991. Fractals and Chaos, New York, Springer-Verlag.
- [3] Barnsley, M.F., and Demko, S. 1985. Iterated function systems and the global construction of Fractals, Proc. Roy. Soc. London, A399, 243-275.
- [4] Jacquin, E. 1992. Image coding based on a Fractal theory of iterated contractive image transformations, IEEE Trans. Signal Process., 1(Jan. 1192), 18-30.
- [5] Jacobs, E.W., Fisher, Y., and Boss, R.D. 1992. Image compression: A study of the iterated transform method, IEEE Trans. Signal Process., 29(Mar. 1992), 251-263.
- [6] Barnsley, M.F. 1990. Methods and apparatus for image compression by iterated function systems. US patent No. 4,941,193.
- [7] Graf, S. 1992. Barnsley's scheme for the Fractal encoding of images. Journal of Complexity. 8 (Aug. 1992), 72-78.
- [8] Barnsley, M.F., and Sloan, A.D. 1998. A better way to compress images. BYTE magazine, (1988), 215-233.
- [9] Fisher, Y. 1994. Fractal Image Compression, Theory and Application. New York, Springer-Verlag.
- [10] Wohlberg, B., and Jager, G.de. 1999. A Review of the Fractal Image Coding Literature. IEEE Trans. Image Process. 8 (Dec. 1999), 1716-1729.
- [11] Polvere, M., and Nappi, M. 2000. Speed-Up In Fractal Image Coding: Comparison of Methods. IEEE Trans. Image Process. 9 (Jun. 2000), 1002-1009.
- [12] Wang, Z., Zhang, D., and Yu, Y. 2000. Hybrid image coding Based on partial fractal mapping. Signal processing: Image communication. 15 (2000), 767-779.
- [13] C. K. Lee and W. K. Lee. 1998. Fast fractal image block coding based on local variances. IEEE Trans. Image Process. 7 (Jun. 1998), 888-891.
- [14] C. M. Lai, K. M. Lam and W. C. Siu. 2003. A fast fractal image coding base on kick-out and zero contrast conditions. IEEE Trans. Image Process. 12 (Nov. 2003), 1398-1403.
- [15] C. M. L T.K. Truong, J.H. Jeng, I.S. Reed, P.C. Lee and A.Q. Li. 2000. A fast encoding algorithm for fractal image compression using the DCT inner product. IEEE Trans. Image Process. 9 (Apr. 2000), 529-535.