Journal of University of Science and Technology Beijing
Volume 9, Number 3, June 2002, Page 228

Fractal image compression based on fuzzy theory

Berthe Kya and Yang Yang

Information Engineering School, University of Science and Technology Beijing, Beijing 100083, China
(Received 2001-11-13)

Abstract: Though progress has been made in fractal compression techniques, the long encoding times still remain the main drawback of this technique, which results from the need of performing a large number of range-domain matches. The total encoding time is the sum of the time required to perform each match. In order to make this method more efficient in practical use, the fuzzy theory based on feature extraction of the projection and normalized codebook method has been provided to optimize the encoding time, based on the c-means clustering approach. The results of the implementation of Rate Mean Square (RMS), Peak signal noise ratio (PSNR) and the encoding time of this proposed method have been compared to other methods like the Feature Extraction and Self-orgarnization methods to show its efficiency.

Key words: fractal compression; fractal optimization; fuzzy logic; c-mean clustering algorithm; hybrid fractal-fuzzy compression

I Introduction

The essence of fractal compression process is the comparison between each range block and domain block, under an affine transformation [1,2]. Basically, a fractal code consists of three ingredients: a partitioning of the image region into portions R_k called ranges and D_k called domains (which may be overlapped). For each domain-range pair two transformations, a geometric one, u_k: $D_k \rightarrow R_k$, which maps the domain to the range, and an affine transformation, v_k, that adjust the intensity values in the domain to those in the range are applied. The collection of transformations may act on an arbitrary image, g, producing an output image T_g. Which is like a collage of modified copies of the domains image g. The iteration of the image operator T is the decoding step, *i.e.*, it yields a sequence of images which converges to an approximation of the encoded image.

The time consuming part of the encoding step is the search for an appropriate domain for each range. During encoding a large pool of image subsets, the domain pool has to be searched repeatedly many times, which by far dominates all other computations in the encoding process. The number of possible domains that theoretically may serve as candidates is prohibitively large. Thus, one must impose certain restrictions in the specification of the allowable domains. In a simple implementation one might consider as domains, *e.g.*, only sub-squares of a limited number of sizes and positions. This defines the so-called domain pool. For a given range, R_k and a domain D_k, the transformations u_k and v_k are constructed such that when the domain image portion is mapped into the range the result $v_k f u_k^{-1}(x)$ for $x \in R_k$ matches the original image f as much as possible. This step (called collage coding) uses the well-known least squares method. From all domains in the pool, select the best one, *i.e.*, the domain D_k that yields the best least squares approximation of the original image as a collage of transformed pieces of itself, which can be viewed as a collection of self-similarity properties. The better the collage fits the given image the higher the fidelity of the resulting decoded image.

The problem of time complexity in fractal image compression was already very clear right from the beginning [3,4]. The earliest, and the latest, implementations use the concept of classification as a tool for complexity reduction. The classification of ranges and domains serves the purpose of reducing the number of domains in the domain pool which need to be considered as a partner for a given range. Previous attempts to reduce the computation times employ classification schemes for the domains based on image features such as edges or bright spots [5, 6]. Thus, in each search only domains from a particular class need to be examined.

The main idea to improving the encoding time described in this paper is base on fuzzy method of the projection and normalized codebook using feature extraction method [5]. The balance of this paper is the follows: the next section focused the Least Square Error based on projection. The section two also focused the problem of Nearest Neighbor search optimization. In the same section, show that the fundamental searching

Corresponding author:Berthe Kya **E-mail**:Kya_abrahamb@hotmail.com

for optimal domain-range pairs is equivalent to solving nearest-neighbor problems in a suitable Euclidean space of feature vectors of domains and ranges. The data points are give by the feature vectors (also called multi-dimensional keys) of the domains, while the query point is defined as the feature vector of a given range. The section three describes the fuzzy theory and the problem relative of the fuzzyfication on image application. In section four the fuzzy algorithm has been given, at the last the conclusion is presented in section 5.

2 Researching of the least squares error nearest neighbor

For the discussion in the paper, assume that an image is partitioned into non-overlapping square blocks of size $N\times N$ called range blocks. This is not a restriction since it will be clear how the principles described carry over to more general partitions.

Consider each range block as a vector $\boldsymbol{R}$ in the linear vector space R^n where $n=N\times N$. The conversion from a square subimage of side length N to a vector of length $n=N^2$ can be accomplished, *e.g.*, by scanning the block line by line. Working with vectors in place of 2D-arrays simplifies the notation considerably without losing generality.

In the encoding process for a range block a search through the codebook blocks is required. The vector representing a codebook block will be denote by $\boldsymbol{D}$, a small set of $p<n$ blocks independent from the image is also considered. Represent them by the vectors $\boldsymbol{B}_1$, $\boldsymbol{B}_2$, $\cdots$, $\boldsymbol{B}_p\in R^n$ which are chosen so as to form an orthonormal basis of a p-dimensional subspace of R^n. The encoding problem can then be stated as the least square problem [5].

$$\mathrm{E}(\boldsymbol{D},\boldsymbol{R})=\min_{a,b_i,\ b_i\in\boldsymbol{R}}\|\boldsymbol{R}-(a\boldsymbol{D}+\sum_{k=1}^{p}b_k\boldsymbol{B}_k)\|=\min_{x\in\boldsymbol{R}^{n+1}}\|\boldsymbol{R}-A\boldsymbol{x}\| \quad (1)$$

where A is an n by $p+1$ matrix whose columns are $\boldsymbol{D}$, $\boldsymbol{B}_1, \boldsymbol{B}_2, \cdots, \boldsymbol{B}_p$ and $\boldsymbol{x}=(a, b_1, \cdots, b_p)\in R^{p+1}$ is a vector of coefficients. This problem should be solved for all codebook blocks $\boldsymbol{D}$ and the one which gives the smallest error $\|\boldsymbol{R}-(a\boldsymbol{D}+\sum_{k=1}^{p}b_k\boldsymbol{B}_k)\|$ is selected on condition that the value of the scaling factor a for the codebook block $\boldsymbol{D}$ ensures the convergence of the decoding process (*e.g.*, by requiring $|a|<1$). This condition can be removed when one uses the orthogonalized representation of ϕien [1]. Dietmar Saupes [5] proved that the solution for equation is:

$$E(\boldsymbol{D},\boldsymbol{B})=\langle\boldsymbol{R},\phi(\boldsymbol{R})\rangle\sqrt{1-(\phi(\boldsymbol{D}),\phi(\boldsymbol{R}))^2} \quad (2)$$

where ϕ is the operator define by $\phi(Z)=\dfrac{OZ}{\|\overline{OZ}\|}$ for $Z=(z_1, z_2, \cdots, z_n)\in R^n\backslash\boldsymbol{B}$.

Thus, the minimization of error $\mathrm{E}(\boldsymbol{D},\boldsymbol{R})$ among domain codebook blocks $\boldsymbol{D}$ can be achieved using an angle criterion: the minimum of $\mathrm{E}(\boldsymbol{D},\boldsymbol{R})$ occurs when the squared inner product $(\phi(\boldsymbol{D}),\phi(\boldsymbol{R}))^2$ is maximal. Since $(\phi(\boldsymbol{D}),\phi(\boldsymbol{R}))^2=\cos^2\angle(\phi(\boldsymbol{D}),\phi(\boldsymbol{R}))$ this means minimizing the angle $\angle(\phi(\boldsymbol{D}),\phi(\boldsymbol{R}))$, or, equivalently to $\angle(OD,OR)$. For the encoding process, look only for the range and domains block whose vector are colinear.

According the theorem 6.1 [5] the minimization of the least squares errors $\mathrm{E}(D_i,\boldsymbol{R})$ for $i=1, \cdots, N_D$ is equivalent to the minimization of the distance between rages and domain blocks expressed by $\Delta(D_i,\boldsymbol{R})$. Where $\Delta(\boldsymbol{D},\boldsymbol{R})=\min(\|\phi(\boldsymbol{R})\pm\phi(\boldsymbol{D})\|)$ the least square error is give by

$$\mathrm{E}(\boldsymbol{D},\boldsymbol{R})=(\boldsymbol{R},\phi(\boldsymbol{R}))g(\Delta(\boldsymbol{D},\boldsymbol{R})) \quad (3)$$

with

$$g(\Delta)=\Delta\sqrt{1-\frac{\Delta^2}{4}}\ \text{and}\ 0\le\Delta\sqrt{2} \quad (4)$$

Before turn to practical issues, use the result of the theorem 6.1 in order to identify all codebook blocks D_i that satisfy a given tolerance criterion

$$\mathrm{E}(\boldsymbol{R},D_i)\le\delta \quad (5)$$

In other words, solving the equality for Δ in the expression for the error $\mathrm{E}(\boldsymbol{D},\boldsymbol{R})$ in the theorem yields a necessary and sufficient conditions for a codebook block $\boldsymbol{D}$ to fulfill the tolerance criterion, which is insured by the corollary 6.1 [5]. Using the consequence of the theorem 6.1 to the corollary 6.1, have

$$\delta>(\boldsymbol{R},\phi(\boldsymbol{R})) \quad (6)$$

These remarks will be useful on the fuzzy parametrization, which can be discussed in the later.

3 Fuzzy theory

3.1 Introduction

Since the introduction of the theory of fuzzy sets by Zadeh [7]. The fuzzy theory has became a powerful framework for image processing, encompassing both high-level computer vision [8,9], and several works has been done in the field.

In fuzzy theory literatures two steps are generally employed to determine the fuzzy set: the identification of a structure and the optimization of the parameters. The methodology to perform these two steps use three phases approach to construct a fuzzy system: Phase 1 outlines the membership functions and system rules for a specific structure, starting from a very simple initial topology. Phase 2 outlines a new and more suitable top-

ology with the information received from the previous step, it determines for which variable the number of fuzzy sets can be used to discretize the domain and where these new fuzzy sets should be located. This, in turn, decides in a dynamic way in which part of the input space the number of fuzzy rules should be increased. Phase 3 selects from the different structures the one providing the best compromise between the accuracy of the approximation. In the approach of this paper, reduce these tree steps in two steps: at first, it is necessary to establish the structure or topology o f the fuzzy system to be used. In this paper because the heuristically membership functions do not reflect the exact [10,11] data distribution in the image; to build the membership functions from the data available, use a clustering technique to partition the data, and then produce membership functions from the resulting clusters. Secondly, the parameters defining the fuzzy system are established whose are been focussed in the next section.

3.2 Fuzzyfication

All literature on fractal image compression has focused on three basic problems. The first problem is to determine a family of contraction maps that can be used to code the image [12]. The second problem is to find the fast and effective contraction map algorithms associated to a given image of which the image is an approximate fixed point [1,13]. The third problem is to analyze the convergence properties of various families of maps and to establish error bounds for decoded image [14,15]. These three problems can be resume in two approaches: encoding and decoding.

Referring to section two some result not least important allows to fixe the fuzzy parameter that is the value of $g(\Delta)$ who belong to interval . For its fuzzyfication it has been proved that by several authors [4] that more the rules are most complex and increase will be the algorithm and the cost. So using the triangular shapes [11], for each above fuzzy parameters, defined only three values [g_{max}, g_{medium}, g_{min}], respective for the following values of If

$$\begin{cases} \Delta\in\left[\sqrt{2},\dfrac{\sqrt{2}}{2}\right], g(\Delta)=g_{max} \\ \Delta\in\left[\dfrac{\sqrt{2}}{2},\dfrac{\sqrt{2}}{4}\right], g(\Delta)=g_{medium} \\ \Delta\in\left[\dfrac{\sqrt{2}}{4},0\right], g(\Delta)=g_{min} \end{cases} \tag{7}$$

where g_{max}, g_{medium}, g_{min} are the values of the middle value the above respective interval.

The value $(\boldsymbol{R},\phi(\boldsymbol{R}))$ depends of the range size whose is fixed according the consideration of the memory storage. The value of δ is compute below.

(1) Compute the feature tolerance δ.

Compute the feature of range f^R by

$$f^R=\sum_{i=1}^{N} f^R[i] \tag{8}$$

(a) Get the minimum and the maximum of feature for each range by

$$\begin{cases} f^R_{max}=\max\limits_{j=1,\cdots,8} f^R_j \\ f^R_{min}=\min\limits_{j=1,\cdots,8} f^R_j \end{cases} \tag{9}$$

(b) Get the feature tolerance of range block for the all image by

$$\delta=\delta_R=\frac{1}{N_R}\sum_{i=1}^{N_R}(f^R_{max}-f^R_{min}) \tag{10}$$

(2) Fuzzy algorithm.

Before give the fuzzy algorithm, practical consideration of memory storage must be take care. In practice, there is a limit in terms of storage for the feature vectors of domains and ranges. For example, the keys for ranges of size of 8 by 8 pixels require 64 floating point numbers each. Thus, 32 K domains from a domain pool would already fill 8 MB of memory on a typical workstation, while like to work with pools of a hundred thousand and more domains. To cope with this difficulty, settle for a compromise and proceed as follows. Down-filter all ranges and domains to some prescribed dimension of moderate size, *e.g.*, d=4×4 =16. Moreover, each of the d components of a feature vector is quantized (8 bits/component suffice).

Use the approach of pixel averaging in order to reduce the dimensionality of the domains and ranges (64 and higher is typical), which is more feasible (here d= 16) and may be improved by better concentrating relevant subimage information in the d components.

Because of the downfiltering and the quantization of both the feature vectors and the coefficients a, b_1, $\cdots$, b_p, it can happen that the nearest neighbor in feature vector space is not the codebook block with the minimum least squares error using quantized coefficients. Moreover, it could yield a scaling factor a being too large to be allowed. To take that into consideration, search the codebook not only for all the nearest neighbor of whose vector is colinear of the candidate range vector. From this set of neighbors the non-admissible domains are discarded and the remaining domains are compared using the ordinary least squares approach. This also takes care of the problem from the previous remark, namely that the estimate by the theorem is only approximate.

(3) Fuzzy parameters.

For this system definite the following fuzzy parameters:

(a) Compute the feature of the 8 isometrics candidate range block, and get the minimum using the following formulas

$$f_{\min}^{R} = \min_{i=1,\ldots,8} \sum_{j=1}^{N_i} f_{ij}^{R}[j] \tag{11}$$

(The number of feature is fixed to five in this paper).

(b) For all domains block whose vector satisfy the colinearity with range block candidate, and compute the 8 isometrics transformations and find the minimum distance by

$$d = \min_{i=1,\ldots,N_i}\left(\min_{i=1,\ldots,8}\left(\sqrt{(f_{\min}^{R} - \sum_{k=1}^{N_i} f_{ij}^{D}[k])^2}\right)\right) \tag{12}$$

(c) If

$$d \le \delta,\ E(D_i, R_i) = d \tag{13}$$

(d) Else compute $\varepsilon = \langle R_i, \phi(R_i)\rangle \times f_{\min}^{D}$ where

$$f_{\min}^{D} = \min_{i=1,\ldots,8} \sum_{j=1}^{N_i} f_{ij}^{D}[j] \tag{14}$$

(i) If $\varepsilon < \frac{\delta}{4}$, $E(D_i, R_i) = \varepsilon \times k_{\max}$.

(ii) If $\frac{\delta}{4} \le \varepsilon < \frac{\delta}{2}$, $E(D_i, R_i) = \varepsilon \times k_{\text{medium}}$.

(iii) If $\frac{\delta}{2} \le \varepsilon < \delta$, $E(D_i, R_i) = \varepsilon \times k_{\max}/3$.

(iv) If $\varepsilon < \delta$, $E(D_i, R_i) = \varepsilon \times k_{\min}$.

(e) If the number of the range block is over break, else got to one.

4 Implementation results

For all implementation, use the Fisher's adaptive quadtree algorithm [6]. The image region is subdivided into squares using a quadtree data structure. The leaf nodes of the tree correspond to the ranges while the domain pool for a given range consists of image subsquares of twice the size.

The **table 1** shows the results of this implementation, the values has been obtain after three iterations of the decoding image. The measure of the quality of the decoding image has been measured using two methods: the root mean square (rms) and the peak signal to-noise-ratio (PSNR) which are definite by:

$$\text{rms} = \sqrt{\frac{1}{N_{\text{rows}} N_{\text{cols}}} \sum_{i=1}^{N_{\text{rows}}} \sum_{j=1}^{N_{\text{cols}}} |p_{ij}^{o} - p_{ij}^{d}|^2} \tag{15}$$

$$\text{PSNR} = 20\log_{10}\left(\frac{M_{\text{graylevel}}}{\text{rms}}\right) \tag{16}$$

where N_{rows} and N_{cols} are the number of rows and columns, respectively, $M_{\text{graylevel}}$ is the maximum gray level value, p_{ij}^{o} is the pixel value of the original image at row i, columns j, and p_{ij}^{d} is the decoded image pixel value.

On the table 1 it can be seen that the fuzzy method give an efficient PSNR comparing to the two other methods: feature extraction and self-organization methods. Particularly comparing the time speed up in the column of the feature extraction, remark an significant increase of the encoding time except the Leave image for the Pool size equal to 16 the speed is decrease. The reason is the complexity of the texture of this image.

In most cases the speed is not change comparing the fuzzy method and the Self-organization method, in some cases the speed is decrease. This reason is that the Self-organization method used in this paper is the Dietmar Saupe method, which is an efficient tool of classification too.

Table 1 results of this implementation

Image size	Pool Size	Feature extraction method			Self-organization method			Fuzzy method and comparison with other methods						
								Fuzzy method			Feature extraction		Self-organization	
		rms ratio	PSNR/ dB	Time/ h:min	rms ration	PSNR/ dB	Time/ h:min	rms ration	PSNR/ dB	Time/ h:min	Change/ dB	Speed up	Change/ dB	Speed up
Lena	16	7.2228	18.8324	19:52	6.7357	19.5445	15:09	9.0667	16.3797	17:49	−2.4527	1.1151	−3.1648	0.9626
	8	19.5048	18.5375	16:00	5.6150	21.9566	6:00	7.0959	18.3263	5:00	−0.2112	3.2	−3.6303	1.2
	1	4.3562	22.1936	8:00	5.4458	22.0181	4:00	6.3217	20.2756	4:00	−1.9180	2	−1.7425	none
Leave	16	16.9275	11.8791	22:09	12.3420	14.3470	16:90	19.134	10.6288	21:59	−1.2503	0.9925	−3.7182	0.7346
	8	8.4451	18.3624	36:00	6.2763	19.7655	17:00	8.0992	17.2478	28:00	−1.1146	1.2857	−2.5177	0.6071
	1	9.3841	17.6738	8:00	7.4084	18.9044	6:00	7.8218	16.7605	7:00	−0.9133	1.1428	−2.1439	0.8571
Berries	16	13.5967	14.7351	20:22	14.5167	14.5364	10:57	13.933	13.9984	15:42	−0.7367	12.9745	−0.538	0.6975
	8	7.4792	19.3951	14:00	7.5767	19.1075	8:00	8.5919	18.5096	8:00	−0.8855	1.75	−0.5979	none
	1	7.2391	19.7274	7:00	7.4925	19.5244	3:00	8.2351	18.6862	3:00	−1.0412	2.333	−0.8382	none
Rose	16	6.7933	18.6975	21:20	6.5120	19.8689	16:54	7.1399	17.9134	20:50	−0.7841	1.024	−1.9555	0.8112
	8	4.4334	20.1249	19:00	4.3027	23.3337	8:00	7.7205	18.3577	8:00	−1.7672	2.1111	−4.976	none
	1	4.4575	22.0245	7:00	4.7152	22.7820	6:00	7.2307	18.5247	6:00	−3.4998	1.1667	−4.2573	none

The advantage of the fuzzy method is that the quality of the decoding image is improved in any case.

The approach reduces the time complexity of the encoding step thereby creating faster fractal image compression. The speed up can be adjusted so that it comes with only minor degradation in image quality and compression ratio or with improvements in both fidelity and compression.

The following result has been obtained by using the computer, which has the following hard resource: the Genuine Intel Pentium II Processor Intel x86 Family 6 Model 8 Stepping 128.0 MB RAM.

The **figure 1** illustrates the rose image with domain size equal to 8. The decoding image has been obtained after 3 iteration.

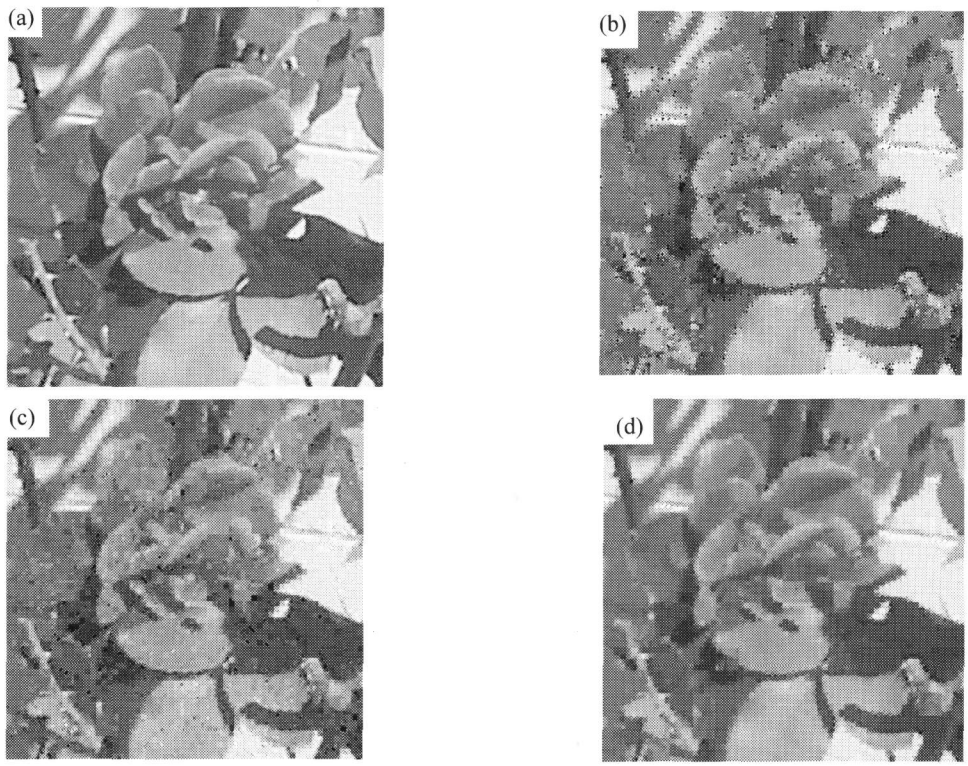

Figure 1 Illustrates the rose image with domain size equal to 8, (a) visualize the original image; (b) the decode image use feature extraction method: *rms* = 4.4334, *PSNR* = 20.1249 and the encoding time is 19 s; (c) the decode image use Self-organization method: *rms* = 4.302 7, *PSNR* = 23.333 7 and the encode time is 8 s; (d) the decode image use the fuzzy method: *rms* = 7.7205, *PSNR* = 18.357 7 and the encoding time is 8 s.

References

[1] S. Lepsoy, G. ϕien, and T. Ramstad, Attractor image compression with a fast non-iteractive decoding algorithm, [in] *IEEE International Conference on Accoustics, Speech, and Signal Processing* [C], Minncapolis, 5(1993), p337.

[2] Stephen Welstead, *Fractal and Wavelet Image Compression Techniques* [M], SPIE Optical Engineering Press, 1999, p.71.

[3] M.F.Barnsley and L.P.Hurd: *Fractal Image Compression* [M], Springer-Verlags, 1993, p.45.

[4] Y.Fisher, *Fractal Image Compression Theory and Application* [M], Springer-Verlag, 1995, p.55.

[5] Dietmar Saupe, Accelerating fractal image compression by multi-dimensional nearest neighbor search, [in] A James, Storer and Martin Cohn, eds., *Data Compression Conference* [C], IEEE Computer Society, 1995, p.222.

[6] Y.Fisher, Fractal compression with quadtrees [J], *IEEE Transaction System*, 15(1995), p.269.

[7] L. A. Zadeh, Fuzzy sets [J], *Information and Controls*, 8 (1965), p.338.

[8] C. De Vleeschouwer, X. Marichal, T. Delmot and B. Macq, A fuzzy logic system able to detect intersting areas of a video sequence [J], *Journal of SPIE*, 3016(1997), p.234.

[9] J.C.Bezdek and S.K.Pal, *Fuzzy Models for Pattern Recognition* [M], IEEE Press, New York, p.345.

[10] K.Hirota and W.Pedrycz, Fuzzy relational compression [J], *IEEE Transaction Systems*, 29(1999), p.407.

[11] Yuval Fisher, *Fractal Image Encoding and Analysis* [M], Springer Published cooperation with NATO Scientific Affairs Division, 1998, p.95.

[12] B. Hurtgen and T. Hain, On the convergence of fractal transform[J], *Processing ICASSP*, 5(1994), p.561.

[13] John Kominek, *Convergence of Fractal Encoded Images* [M], IEEE Computer Society, 1995, p.242.

[14] L. X. Wang and J. M. mendel, Generating fuzzy rules by learning from examples [J], *IEEE Transaction Systems*, 22 (1992), No. 6, p.1414.

[15] M.S.Lazar and L. T. Bruton, *Efficient Fractal Block Coding Using Multiresolution Decomposition Based Search Constraints* [R], Computer Engineering Technical Report, p. 1235.