Fractal Image Compression Using Neural Network

Prof.Dr. Abdel-Razek A. Abouel-Nour[1] , *Prof.Dr.* A.Ali Fahmy[2],
Dr. Abdel-Latief H. Abo Ali[3], Eng. *Sabry M.* Abdel-Moetty[4]

[1] Ind.Eng*: and Systems Eng. Dept., Faculty of Engineering, Zagazig University,Zagazig, Egypt*
[2] *Computer and Systems Eng. Dept., Faculty of Computer and I.S, Cairo University,Cairo, Egypt*
[3] *Computer and Systems Eng. Dept.; Researches & Tech. Development, Egyptian Air Force, Egypt*
[4] *Computer and Systems Eng. Dept, Integrated Logistic Systems, Egyptian Air Force, Egypt*

Abstract

In this paper, a novel fractal image compression technique using neural network approach. The proposed algorithm uses neural network (Self organizing method) to select the proper set of domain blocks for a selected range block. It also efficiently uses the neural network to identify the proper orientation. The algorithm remarkably reduces the compression time without suffering of image distortion. The algorithm uses the RAW image data without going through feature extraction.

Keywords: Image Compression, Artificial Neural Network (ANN), Iterated Function System (IFS), and Partitioned Iterated Function System (PIFS), Self organizing method (SOM)

Introduction

Fractal image compression is a promising new technology [2]. The birth of this technology could be traced to the book "The Fractal Geometry of Nature" [1]. The book put forth the guidelines of a new geometry, the fractal geometry that could be employed to generate natural looking images. The work of mathematicians continued [1] [2] [3]. These studies generated basically the iterated function system, IFS, and the collage theorem. The theorem presented the conditions in an iterated function system to represent an image.

The work of mathematicians for generating a natural looking image raised the inverse problem. The key for solving such a problem was to search on self-similarities. Initially, the search was to find self-similarities between the image as a whole and its parts [3] [4]. This approach is found to be unrealistic [14]. The realistic approach was based on finding self-similarities between large parts and small parts of the image, i.e. The image will be mapped to a set of relations between large blocks called domain blocks and small blocks called range blocks.

In this paper, we will present an algorithm for Fractal image compression based on self-organizing method classification ability. To make the problem controllable, we will limit ourselves to the gray scale images. Before proceeding in describing our algorithm, it will be beneficial to introduce briefly the partitioned iterated function transformation and the self-organizing neural network. For more details please refer to [13] [15] [16] [17].

The partitioned iterated function transformation.

In this paper to be more specific, we will rely on Jacquin assumptions [3]: -

1- The domain and range blocks are assumed to be square.
2- The size of the domain block is restricted to twice the size of the range block.
3- The isometric operations allowed are the 8 (out of 81 possible).
4- The symmetry operations are rotations by (0, 90, 180, and 270) flipping against the horizontal and the vertical median, and flipping against the reverse and the forward diagonals.

5- The relation intended, the affine transformation, is a linear relation.

In fractal image compression algorithm, the image to be compressed is divided into two sets of image blocks the range blocks set and the domain blocks set. The image range blocks set contains all the non overlapped blocks of some arbitrary size. The image domain blocks set contains all the possible non-duplicate blocks of size equal to range blocks.

The fractal image compression seeks an affined transform $w = F(e, f, m, o, s)$ for each range block. Where:

e, f : Position of the domain block of best match.
m : Isometric operation.
o : Offset
s : Scaling factor.

o Offset parameter can be quickly computed as the mean of the range block.

The rest of the parameters require optimal pairing between range and domain blocks. This optimal pairing requires seek of each range block in the domain blocks pool. The optimality requires a metric measure. Using the root mean square as our metric and assuming that D is the optimal domain block for a range block R ,then the scaling factor is [3]: -

$$s = \frac{\sum_i \sum_j r_{ij}^o d_{ij}^o}{\sum_i \sum_j \left(d_{ij}^o\right)^2}$$

Where d_{ij} are the elements of the domain block after scaling down to the size of the range block and transforming it to zero-mean. And r_{ij} is an element of the range block R after transforming to zero-mean.

The decompression process begins with a seed image. The transforms are applied on the seed image to produce a new seed of the next iteration. The process is repeated until it converges or a preset number of iterations are achieved. The decompression process is found to be convergent within a few numbers of iterations under the restriction:

$$d(w(P_1), w(P_2)) < s\, d(P_1, P_2) \qquad [1]$$

Where $0 < s < 1$. This formula shows that the application of contractive map always brings points closer together (by some factor less than 1).

The major issue under all these assumptions is the seek strategy for finding the best pairing. The problem computational complexity is in the order of $O(n^2)$ i.e. when the size of the image n grows the computational demands becomes time consuming. Figure 1 shows the computational demands for a single comparison under an exhaustive search in function of image size.

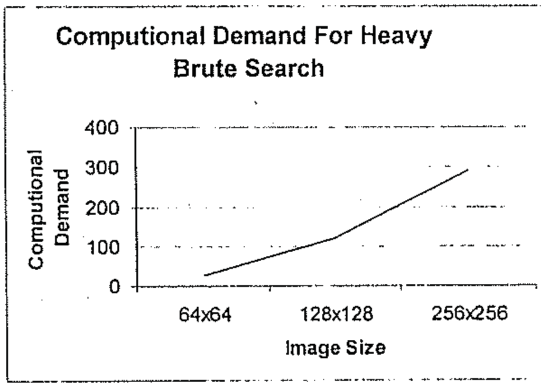


Figure (1) Computational Demands of the Exhaustive search

The search problem led to a lot of research [3] [7]. Hence Figure 2 draws the relationship between the number of operations required in accordance to image size, assuming Block Size 8x8.The *"Light brute"* search algorithm considers only even pixel location domains. The *"restricted area"* search algorithm restricts the search pool to narrow neighborhood. The *"local spiral" search* defines a search path starting from the range block position. The *"categorized search"* is based upon a preprocessing of the domain and range blocks i.e. the pool is limited to the domain pool of the same category and the categorization approach is based mainly on the feature extraction.

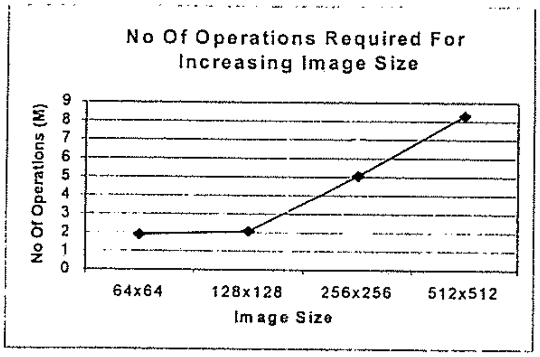


Figure (2) No Of Operation Required As The Image Size Increases

In this paper we will propose a neural based search strategy that uses the RAW image and block data to determine the proper domains as well as the proper orientation. The neural network technique used is the self-organizing map neural network. In the next section, we will briefly describe the technique used [13] [15].

The Self Organizing Feature Map

The self-organizing feature map SOFM, is an unsupervised artificial Neural Network technique [13][16]. The overall network mapping function converts an arbitrary dimensional pattern into responses of one or two-dimensional array of neurons. The mapping function is done using a codebook. Neuron weights are evolved during the network training process. The network structure is a two-layer of neurons: input layer and output layer. The input neurons and the input patterns are dimensionally equal. The output layer contains array of competitive type of neurons. i.e.

Let us define the set of input patterns to be processed by the network:

$X = \{x_1, x_2, x_3, ...\}, \quad X \subset R^n$

The network has an output layer neurons in form of rectangular set:

$\beta = \{N_{11}, N_{12}, N_{13}, ..., N_{1c}, N_{21}, N_{22}, ..., N_{rc}\}$

of dimensionality $[r \times c]$.

The codebook evolved during the training process, states the neuron weights as

$$\varpi = \{W_{11}, ..., W_{1c}, W_{22}, ..., W_{rc}\} \quad and\, \varpi \subset R^n \qquad [2]$$

The network response to an input vector y_k is as follows:

The neurons of the network react in parallel to the input by computing

$r_{ij} = \Gamma(W_{ij}, y_k)$, Where Γ is some similarity distance metric in R^n.

A network neuron N_{mn} will be declared a winner if and only if

$$r_{mn} = \min\{r_{ij}\}, \quad 0 < i < r+1 \quad 0 < j < c+1 \qquad [3]$$

The network has two operational modes: the learning mode and the recall mode. The network is started with some arbitrary codebook, normally random. Then it is set to the training mode to generate a suitable codebook using the selected training set. After convergence of the training process the network can bet set to the recall mode.

During the training and as the training goes on the time t, the weight vectors of a winning neuron and its neighborhood are incremented as follows: $\Delta W_{ij} = \alpha(d_{ij}^{mn}, t)(y_t - W_{ij})$.

The function α is reciprocally proportional to time and the distance from the winning neuron d_{ij}^{mn}

In the recall mode, the neurons will compute their responses and the winner will be declared and assigned a winning response y_{mn} for example "True", the rest of the neurons are assigned a lose response for example "False". The winning neuron class is included in the input pattern class.

Neural based fractal image compression algorithm.

Let n be the range block length. $\Omega = \{D_1, D_2, ...D_{dn}\}, \quad \Omega \subset R^{2n} x R^{2n}$ Be the set of domains blocks and, $\Psi = \{R_1, R_2, ...R_{rn}\}, \quad \Psi \subset R^n x R^n$ be the set of range blocks for an image I. We denote η as a scale down mapping $\eta : R^{2n} x R^{2n} \rightarrow R^n x R^n$, ξ_i denotes the isometric transforms $i = 1,2,...8$ and φ is a zero mean transform over blocks in $R^n x R^n$. The search problem, assuming rms as the similarity measure, can be stated as finding D_s *and* ξ_x for each range block R_m such that

$$\|\xi_i(\varphi(\eta(D_s))) - \varphi(R_m)\| \qquad [4]$$
$$\leq \|\xi_i(\varphi(\eta(D_j))) - \varphi(R_m)\| \quad \forall D_j \in \Omega \quad \textit{and} \quad \xi_i$$

To limit the search to some measure of quality equation [4] can be rewritten as

$$\|\xi_k(\varphi(\eta(D_s))) - \varphi(R_m)\| \leq e$$

where e is acceptable apriori error

Let's assume N denotes the mapping of a trained self-organizing feature map

$N : R^{nxn} \rightarrow i \quad i \in R, \quad 0 < i < r * c \quad i$ Is the class index.

The compression algorithm goes as following

1- Set an index j =*1*.

2- Classify the domain set and form domain subsets

$$\Omega_i = \{D_s / N(\xi_j(\varphi(\eta(D_s)))) = i\} \quad \forall D_s \in \Omega. \qquad [5]$$

3- Set the remaining set of ranges $\psi^j = \Psi$

4- Classify the set of remaining of ranges ψ^j into subsets

$$\psi_i^j = \{R_s / N(\xi_j(\varphi(R_s)) = i\} \quad \forall R_s \in \psi^j \qquad [6]$$

5- For each range class set ψ_i^j the search is constrained to the domains Subset Ω_i. That is for a range block $R_m \in \psi_i^j$ the best matched domain block D_s is

$$\|\xi j(\varphi(\eta(D_s))) - \varphi(R_m)\| \leq \|\xi j(\varphi(\eta(D_k))) - \varphi(R_m)\| \forall D_k \in \Omega i$$

The rms error and the affine transform are computed.

6- The rms is computed for each range block and compared to the previously computed ones.

7- Update the transforms for all blocks have smaller rms.

8- mark all the blocks that satisfy $\|\xi_k(\varphi(\eta(D_s))) - \varphi(R_m)\| \leq e$

9- Set $\psi_i^{j+1} = \psi_i^j - \textit{marked blocks}$

10- Increment j and repeat from step 4 until j reaches *9 or* ψ_i^j is empty.

One of the features of the SOFM is: Distance between the trained neurons is meaningful i.e. the distance means closeness in features. This property is employed to reset Ω_i to the closest neighbor set when passed empty from step five.

By the end of the algorithm the set ψ_i^j may not be empty however the best mapping is associated. These blocks could be either border elements of their sets or the measure of quality used is not realistic. The first problem can be avoided easily if we repeated the algorithm with ψ^9 and considering the search pool is the nearest neighbor. Otherwise the blocks could be broken into smaller sizes [2].

Test and Results

We trained neural networks of sizes 4x4, 8x8, 16x16 and 32x32 using "Linna" standard image. The computational demands of the heavy brute search and the proposed algorithm are presented in Figure (3). The Figure shows:

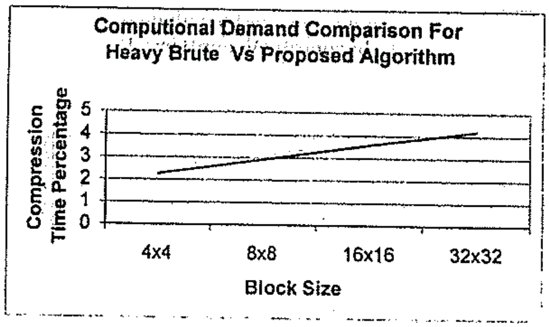
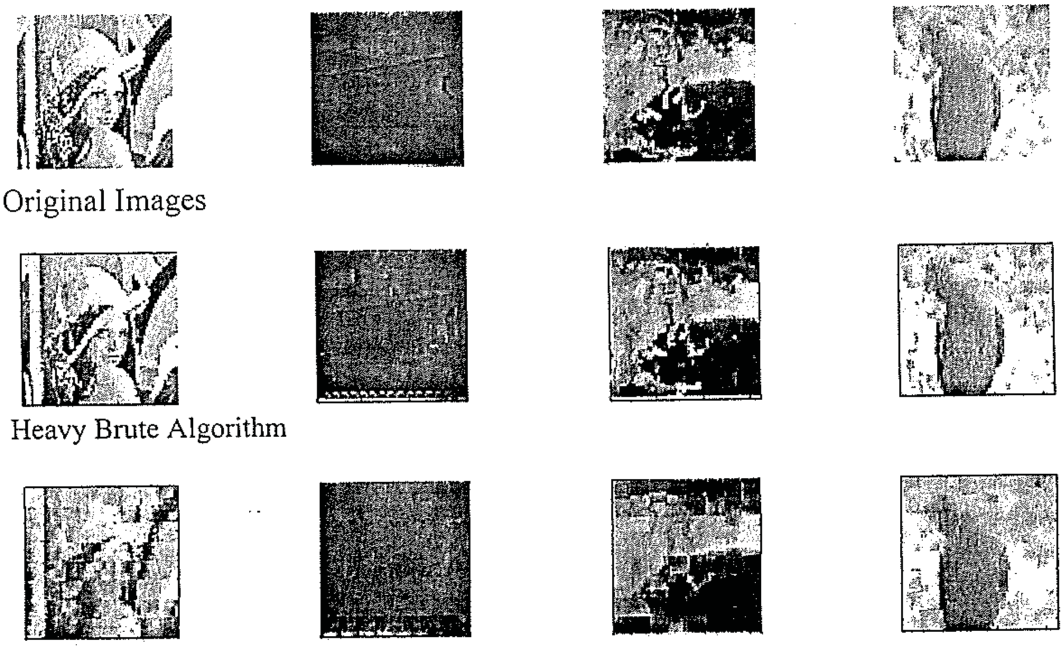


Figure (3) Comparison between the Computational Demands for Heavy Brute And Novel Algorithm

The algorithm is used in compression of a variety of images. Table (1) shows some of the results on some standard images.

	B.Size	*C.Time%*	*RMS*		*PSNR*	
		Heavy / proposed	*Proposed*	*Heavy Brute*	*Proposed*	*Heavy Brute*
Linna	4	27.8/12.3	0.17	0.12	25.5	27
Road	4	37.5/14.5	0.15	0.11	21.1	27.8
Sea	4	14.5/5.8	0.19	0.12	22.8	26.5
Redbird	4	18.06/9.9	0.13	0.07	26.2	30.5
Linna	8	7.1/ 2.4	0.28	0.25	18.1	20
Road	8	5.9/3.4	0.19	0.14	22.2	24.5
Sea	8	6.3/2.9	0.21	0.17	18.6	21.02
redbird	8	5.9 /3.1	0.26	0.23	21.2	23.45



(a) Original Images

(b) Heavy Brute Algorithm

(b) Proposed Algorithm

Figure (4) Samples of the compressed images

(a) is the original image
(b) After compression and decompression.

Conclusion

The objective is to reduce time complexity during search process by proposing fractal image compression using self-organizing network method.

The proposed algorithm decreased the encoding time while keeping a respectable image fidelity and a reasonable compression ratio.

Comparing the proposed algorithm with the "Heavy brute force" algorithm results assured that speed trades off quality. Heavy brute force algorithm always finds a best match for all domain and range blocks in contrary to the proposed algorithm that does not always guarantees a best match.

References

1. Benoit B. Madelbort. **"The Fractal Geometry Of Nature.**",W.H. Freeman and company, New York, 1977.
2. Yval Fisher. "**Fractal Image Compression.**", SIGGRAPH' 92 Course notes, an Diego Super Computer Center, University of California, 1988.
3. Dietmar Saupe. "**Breaking the time Complexity of Fractal Image Compression.**" Institute für Infomatic, Universität Freiburg, Rheinastr. 10-20,79104 Freiburg, Germany, 1994.
4. Behnam Bani-Eqbal. "**Speeding up Fractal Image Compression**", 1Department of computer Science, university of Manchester, UK, 22 September 1994.
5. A. E. Jacquin, "**A Fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding**" . PHD thesis, Georgia Institute of Technology, aug. 1989.
6. A. E. Jacquin, "**Image coding based on a fractal theory of iterated contractive image transformations,**" IEEE Transactions on Image Processing vol. 1, pp. 18-30, Jan.1992.
7. 7. T. Lookabaugh and R. M. Gray , "**High-resolution quantization theory and the vector quantizer advantage,**" IEEE Trans. on Information Theory,vol. 35,pp. 1020-1033, sept. 1989.
8. M.F. Barnsley , "**Fractals Everywhere**". San Diego, CA.: Academic Press, 1988.
9. J. Stark, "**Iterated function systems as neural networks,**" Neural Networks , vol. 4, pp. 679-690, 1991.
10. J. E. Hutchinson, "**Fractals and self similarity ,**" Indiana Univ. Math. Journal vol.30, no. 5, pp. 713{47, 1981.
11. S.-I. Amari, "**theory of self-organizing neural nets,**" IEEE Transaction Syst. Man and Cybernetics , vol. SMC-13, pp. 741-748,sept. 1983.
12. S. Grossberg, "**Adaptive pattern classification and universal recording: I. parallel development and co ding of neural feature detectors,**" Biological Cybernetics , no. 23, pp. 121-134, 1976.
13. T. Kohonen, "**Self-Organization and Associative Memory**" . Springer Series in
14. Information Sciences, Springer- V erlag, 3 ed., 1989.
15. H. Ritter and K. Schulten, "**Convergence properties of Kohonen's topology conserving maps: Fluctuations, stability, and dimension selection,**" Biological Cybernetics, no. 60, pp. 59{71, 1988.
16. V. Tolat, "**An analysis of Kohonen's self-organizing maps using a system of energy functions,**" Biological Cybernetics , no. 64, pp. 155{164, 1990.
17. T. Kohonen, "**The self-organizing map,**" IEEE Proc. , v ol. 78, pp. 1443-1464,sept. 1990.