Scientific Journal of King Faisal University (Basic and Applied Sciences) Vol. 7 No. 1 1427H (2006)

On Improving Processing Speed of Image Fractal Compression Using Artificial Genetics Techniques

A. Gafour, M.K. Feraoun & A. Rahmoun*

Evolutionary Engineering & Distributed Information Systems Laboratory
University of Sidi Bel Abbes, Algeria
* Faculty of Computer Science, King Faisal University, Saudi Arabia

Abstract :

Artificial Genetics tools have been for decades a subject for intensive investigations. From Genetic Algorithms and Hierarchical GA's to Evolutionary Engineering tools such as Genets, these tools have proven their efficiency in search and optimization problems in multi-dimensional spaces.

Fractal image compression has also been for decades a challenge for researchers facing the emergence of new web based technologies and applications.

Several works have been proposed in the literature of GA-Based Fractal compression emphasizing on Compression rate-Image quality tradeoffs.

In this paper we propose a dynamic Genetic Algorithm to improve performance of Fractal image compression based on local IFS. An adequate GA coding is used to address the different image parameters. The algorithm is set to fit all possible decomposition schemes regardless the range block size and position.

Furthermore, to avoid premature convergence, a linear scaling is applied to each individual fitness, then a roulette wheel method is applied for selection process.

Such modified GA is intended to speed up the coding phase by varying coding period through tuning GA parameter settings according to maximum bloc similarities.

Several computer simulation tests have been performed on Fractal images. Results show a major reduction in processing time during bloc-range search process without major loss of image quality .

Introduction :

The tremendous advances of information technologies as well as the emergence of new web-based technologies have created new needs in dealing with multimedia internet-based information and applications.

Several works in the literature have investigated ways to better compression time and ratio without loss of image quality.

These various research papers emphasize essentially on how genetic algorithm-based techniques handle tradeoffs on compression ration or compression time versus image quality.

In [Alan95] , authors propose a bibliography of GA's in optics and image processing.

[Kim2002] focuses on unsupervised image segmentation using a distributed GA.

In [Xuan96], the author used a modified distributed GA for image segmentation by adding the capabilities of fuzzy systems to overcome contour effects.

[Redmi96] and [Mitra98] propose an improved GA for solving IFS code of fractal images.

[Beret 95] proposes modified and improved GAs for fast search in fractal image coding. These rarely take into consideration constraints on image parameters, image decomposition and segmentation schemes. This had instead proved affecting substantially image overall quality.

Fundamentally, the fractal image compression problem can be defined as : “given a Ranges(I), the set of all ranges blocks obtained by the decomposition schema, how to construct the set of all possible domain blocks Dom(I) such that it exists a transformation T : Dom(I) → Ranges(I) ; T must guaranty $\forall i, \exists j /\ T(Dj) = Ri$ ”. The compressed image is constituted of a set of IFS and Dom (I). A transformation is associated to each Ri, it codes the Dj coordinates and the specific parameters of the transformation T.

The image compression problem put forward three major requirements: speeding up the compression algorithm, improving image quality or increasing compression ratio. The compression ratio depends on the size of Ranges(I).

Fundamentally, the compression algorithm speed depends on the manner of exploring the search space to determine a block domain Dj.

The image quality is not absolutely separable of the previous two criteria.

The main problem of all fractal compression implementations is the execution time. Algorithms can take hours to compress a single image. So, the major variants of the standard algorithm were proposed to speed up computation time. But most of them lead to a bad image quality, or a lower compression ratio.

For example, the Fisher’s proposed classification schema has greatly accelerated the algorithm, but image quality was poor, due to the search space reduction imposed by the classification, witch eliminate a lot of good solutions. Then, the main problem is how to explore all domain blocks presents in the image, not in exhaustive way as standard algorithm does, and without omitting any possible block (solution) as classification schema does.

In this paper we use genetic algorithms (GA) [Dasg2000] to satisfy such goal and to optimize the domain blocks search.

This is feasible by the fact that: GAs act randomly, are implicitly parallel and are directed by the fitness only.

This paper presents essentially the GA implementation. We have also implemented both Bernesly standard algorithm [Bans93] , and Y. Fisher algorithm based on classification schema [Hutch81] , [Saup94], [Venc94], [Shon91].

A performance comparison between different algorithms is also presented. Section 2 specify the GA characteristics, section 3 deal with regular partitioning and QuadTree genetic implementation, tests and results summary.

Standard Schema With Ga :

GAs are used to improve compression schema, principally to accelerate coding time. For each range domain Ri, the set of all possible domain blocks is genetically browsed until we find an appropriate solution.

The GA search space parameters are the domain block coordinates and the isometric flip.

The luminance and contrast (S and O) parameters are computed as done in the standard algorithm.

Chromosome Codification :

According to the search space parameters a chromosome is constituted by three genes : Xdom , Ydom the domain block coordinates and the isometric flip. Xdom, Ydom and flip are integers.

Xdom $\in$ [0, L], L is the image length.
Ydom $\in$ [0, W], W is the image width.
flip $\in$ [0, 7], eight isometric flip.

X_{dom}	Y_{dom}	Flip

Figure. 1. Chromosome representation

The Fitness Function :

The fitness function assigns to each individual in the population a numeric value that determine its quality as a potential solution. The fitness denotes the individual ability to survive and produces offspring. In our case, the fitness is given by the inverse of the RMS error between the coded range block, and the domain block.

Mutation operator modifies the chromosome genes randomly according to the mutation probability. the genes determined by the transformation co-ordinates Xdom and Ydom, and transformed with corresponding luminance and contrast values.

The smaller is the RMS, the better is the image quality. We normalize the fitness to 1 by taking the inverse value of the RMS, so that the quality is best when fitness is close to 1.

The RMS equation and the fitness function and the transformation parameters formulas are given in the following, where ai are domain elements, and bi denote the range elements :

Fitness function (T)= 100 / (RMS(Ri ,T(Ri))).

$$RMS=\frac{1}{n}\left[\sum_{i=1}^{n2} b_i^2+S(S.\sum_{i=1}^{n2} a_i^2-2\sum_{i=1}^{n2} a_i b_i+2.o.\sum_{i=1}^{n2} a_i)+o.(on2-2\sum_{i=1}^{n2} b_i)\right]$$

$$s=\frac{n^2.\left(\sum_{i=1}^{n^2} a_i b_i\right)-\left(\sum_{i=1}^{n^2} a_i\right).\left(\sum_{i=1}^{n^2} b_i\right)}{n^2\sum_{i=1}^{n^2} a_i^2-\left(\sum_{i=1}^{n^2} a_i\right)^2}$$

and

$$o=\frac{1}{n^2}.\left[\sum_{i=1}^{n^2} b_i-S\sum_{i=1}^{n^2} a_i\right]$$

Genetic Operators :

The crossover and mutation operators ensure the production of offspring. These genetic operators must be defined according to the chromosome specification.

Crossover Operator :

The crossover operator combines two individuals (the parents) of the current generation and produces two offspring individuals. According to our chromosome specification new schema of the crossover operator is proposed: the offspering individuals coordinates are obtained by a linear combinaison of the parents coordinates and the offspering flip value is randomely choosen from one of the parents flip value. The new offspring coordinates are computed according to the following formula:

For the first offspring :

$$X_{dom} = a^* X_{dom}^{p1} + (1-a)^* X_{dom}^{p2}$$
$$Y_{dom} = a^* Y_{dom}^{p1} + (1-a)^* Y_{dom}^{p2}$$

For the second offspring:

$$X_{dom} = (1-a)^* X_{dom}^{p1} + (1-a)^* X_{dom}^{p2}$$
$$Y_{dom} = (1-a)^* Y_{dom}^{p1} + (1-a)^* Y_{dom}^{p2}$$

Where the constant “a” is a random number in the interval [0 , 1]

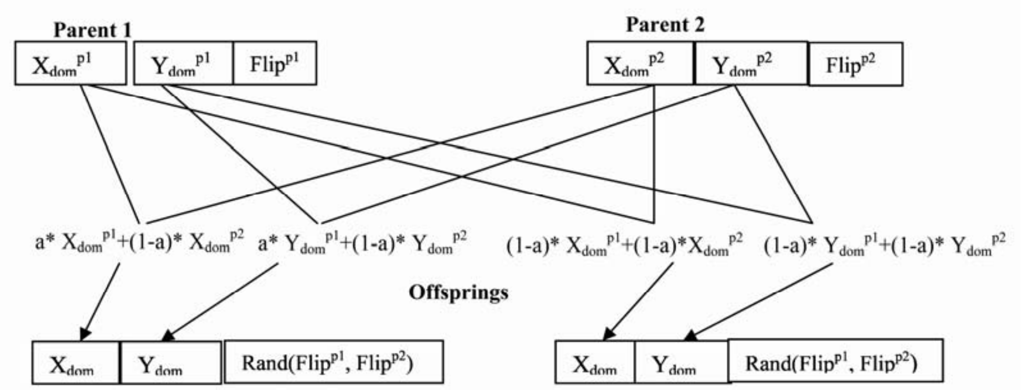

Figure 2. crossover operator schema

Mutation Operator :

Xdom, Ydom and flip are changed with a random generated value respectively in [0, L],[0, W], and [0, 7] intervals. The figure 3. illustrate the mutation operator schema

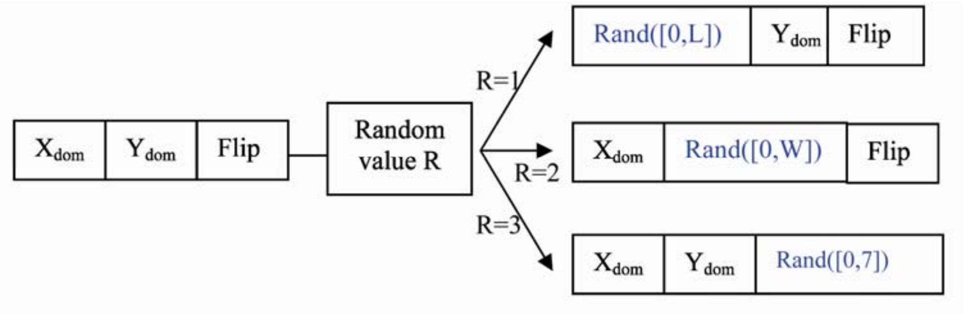

Figure 3. Mutation operator schema

Selection Process :

To avoid the premature convergence effect, linear scaling is applied to each individual fitness. Then, the Roulette wheel method is used as a selection process.

Termination Criteria

When applied to a given range block, two criteria can cause the genetic algorithm termination:

- An individual with an acceptable fitness is found;
- The last generation is reached (either by setting up a finite number of generations or a fitness limit (.99) as criteria to terminate the search algorithm).

```
Genetic-Compression Algorithm (Input I :256x256 gray scale image, Output W: Coded IFS);

    Decompose the input image into range blocks according to the decomposition schema;
    For each block R in Ranges(I) do
                      - Generate a random population of chromosomes;
                      - Compute fitness for all individuals;
        While (No optimal domain block is found) and (last generation is not reached) do
                - Generate a new population:
                              {Use selection process,
                               Apply Crossover and Mutation operators};
                - Compute fitness for all individuals of the new generation;

        End While;
        - Write obtained transformation parameters to the output W;
    End For.
End GCA.
```

Parameters inherent to the genetic algorithm:

- Population size;
- Crossover rate;
- Mutation rate;
- Number of generations.

Parameters inherent to the fractal image compression schema are :

- The range blocks decomposition size (used with regular partitioning);
- The lowest block size used for ranges decomposition (in the case of QuadTree schema);
- The number of flips and isometrics applied to each domain block;
- The decomposition error limit, this parameter is introduced to improve the QuadTree decomposition schema;
- The RMS error limit fixed to decide if a given transformation is accepted.
- The number of bits used to quantify and code luminance and contrast parameters, fixed experimentally to 5 and 7 bits respectively.

Population Size	100
Maximum generations	20
Crossover rate	From 0.7 to 0.8
Mutation rate	0.1
RMS limit	5.0
Decomposition error limit	10.0
Flips and isometrics count	8

Figure 4. Optimal set of parameters

The values of range blocks size and lowest decomposition level depends on the used decomposition schema, they are fixes in the case of regular partitioning. For the QuadTree decomposition, we set the lowest block size to 4x4 pixels to achieve highest reconstruction quality, and to 8x8 to obtain acceptable quality with high compression ratio. In the following, we present the obtained results for different decomposition methods and with different parameters combinations, A comparison with both standard and classification based algorithms is also presented and discussed.

Tests and Results

Genetic Compression Algorithm With Regular Partition

The decomposition schema is a regular partition with 8x8 and 4x4 block size. The genetic algorithm optimises the domain block search. Results are as follow:

8X8 Decomposition

With crossover rate fixed to 0.7, mutation rate to 0.1 and population size to 100. Figure 5 gives results for genetic, standard and classification compression schema applied to Boat image. It is clear that genetic schema greatly reduces compression time without significant loss of image quality. Figures 8 and 9 shows restitution of Boat and Barb images. Figures 6 and 7 show quality and compression ratio variations for different RMS error.

	RMS = 2.0	**RMS = 5.0**	**RMS = 10.0**	**RMS = 15.0**
Standard Algorithm Results	PSNR=24.25 db Time= 35 m 18 s	PSNR=23.95 db Time=28 m35 s	PSNR= 23.12 db Time= 21 m 69 s	PSNR=22.62 Time= 18 m 32 s
Genetic Algorithm Results	PSNR=23.56 db Time=24 s	PSNR=22.93 db Time=16 s	PSNR=22.51 db Time=12 s	PSNR=22.01 db Time=9 s
Classification Algorithm Results	PSNR= 22.11 db Time= 1m 56 s	PSNR=22.11 db Time=1m 09 s	PSNR= 20.01 db Time= 56 s	PSNR= 19.66 db Time= 44 s

Figure5. Results for the three presented algorithms with 8x8 regular partition Applied to Boat image.

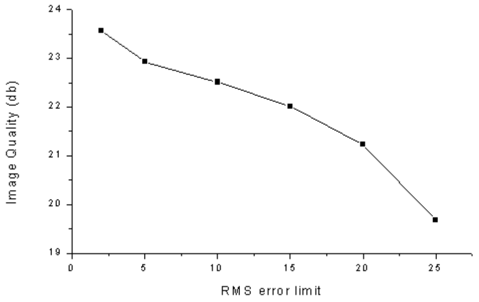

Figure 6. Boat image Quality variation for different RMS error

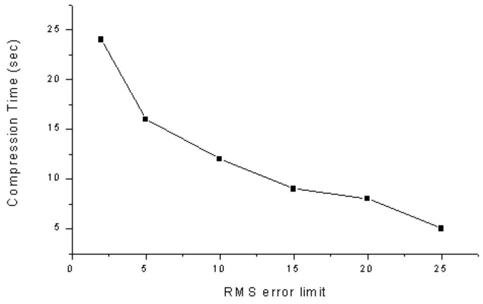

Figure 7. Boat image Compression ratio variation for different RMS error

4x4 Decomposition :

High quality is always obtained in this schema with all proposed algorithms, but the genetic algorithm increases greatly compression speed. The compression ratio is very low. Figures 8 and 9 shows decompressed and original image of Lena with RMS=5.0. Table in figure 10 gives results for genetic, standard and classification compression schema applied to Boat image.

Figure 8- Original Image of the Boat

Figure 9- Decompressed Image with 4X4 decomposition scheme using genetic algorithm

Genetic Algorithm With Quadtree Decomposition :

The QuadTree schema is the best way to decompose compressed image, and to make range blocks suitable to the image content. The major problem is the high computation time. The quality of the decompressed image depends only on RMS error because different size of range blocks can be used according to the current image area. Genetic algorithm, used with this method, improves compression speed. If we decrease the error limit, we obtain good image quality without greatly slowing the compression process: this is the main advantage of our implementation. The following table show different performances with different value of RMS error limit using fixed value for other parameters: population size=100, mutation rate = 0.1, crossover rate = 0.7 and maximum generations count =20. figures 11 and 12 shows that image quality is inversely proportionate to RMS error limit. And compression ratio is proportionate to that value. The compromise value of this parameter is 5.0, it gives very acceptable performances.

RMS Limit	Execution Time	Quality (db)	Compression Ratio	Ranges count
0.0	2 m 44 s	35.66 db	4.29 :1	4069 block
2.0	1 m 56 s	35.03 db	6.35 :1	2770 block
4.0	49 sec	34.89 db	9.28 :1	2023 block 1792
5.0	43 sec	34.80 db	9.82: 1	block
8.0	36 sec	34.5 db	9.95 :1	1768 block
10.0	33 sec	30.5 db	10.05 :1	1750 block
15.0	21 sec	22.33 db	13.66 :1	1288 block
20.0	14 sec	19.36 db	19.34 :1	910 block
25.0	15 se	19.01 db	26.25 :1	670 block

Figure 10. Different compression results of Boat image while applying different values of RMS error limit

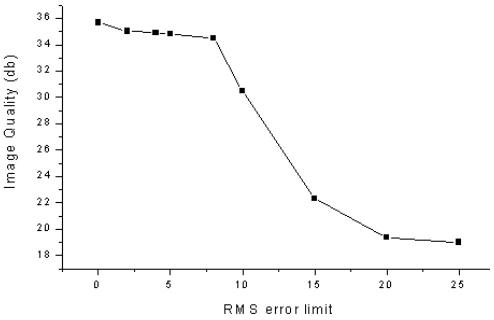

Figure 11. Boat image quality variation according to RMS limit values

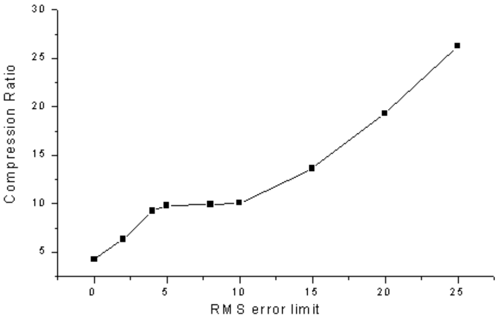

Figure 12. Boat image compression rate variation according to RMS limit values

Figure 13- Original Image of the Boat

Figure 14- Image of the Boat after restitution RMS=5.0 (Ratio 9,14:1)

Image	Standard Algorithm			Genetic Algorithm			Classification Algorithm		
	Rate	Quality	Time	Rate	Quality	Time	Rate	Quality	Time
Boat using 8x8 decomposition schema	17.06:1	23.95 db	28m 35s	17.06:1	22.93 db	16 s	17.06:1	22.91 db	1m 10s
Boat using 4x4 decomposition schema	4.2:1	31.95 db	16m 28s	4.2:1	30.88 db	36 s	4.2:1	31.12 db	1 m
Boat using QuadTree decomposition schema	11.48 :1	32.01 db	1:10:11	9.88 :1	33.25 db	42 s	9.73 :1	30.05 db	2m 55s

Conclusion :

It is clear that the best image quality is always obtained using the standard schema, but its computation time makes it unpractical. So we must accept less quality in favor of quick compression. The Fisher’s approach was proposed to satisfy this constraint. Our main goal was to accelerate standard compression schema, without greatly decreasing both image quality and compression ratio.

The results presented above prove that the genetic fractal compression algorithm seems to be the best. Further more this work demonstrates the genetic algorithm ability to solve complex problems.

References :

1. [Alan95] Jarmo T. Alander. Indexed bibliography of genetic algorithms in optics and image processing. Report 941 -OPTICS, University of Vaasa, Department of Information Technology and Production Economics, 1995. http://citeseer.nj.nec.com/article/alander00indexed.html site (ftp://ftp.uwasa.fi/cs/report94-1/)
2. [Kim2002] - Eun Yi Kim, Se Hyun Park, Sang Won Hwang, Hang Joon Kim, " Video sequence segmentation using genetic algorithms", Volume 23 , Issue 7 May 2002, Elsevier Science Inc. New York, NY, USA
3. [Xuan96] Xuan, Dequn "An improved genetic algorithm of solving IFS code of fractal image". In Proceedings of the 3rd International Conference on Signal Processing, volume 2, pages 1405-1408, Beijing (China), 14.-18. October 1996. IEEE, New York, NY.
4. [Redmi96] Redmill, Bull et al , "Genetic algorithms for fast search in fractal image coding ", Proc.SPIE Int.Society for Optical eng., 1996.
5. [Mitra98] - Mitra, Murthy et al , "Technique for fractal image compression using genetic algori.. " IEEE Trans. on Image Processing, vol. 7; no2, 1998.
6. [Beret95] Beretta, Tettamanzi, "An evolutionary approach to fuzzy image compression ", Proc. of the Italian workshop on Fuzzy Logic, pp 49-57, Naples, Italy, 21-22 Sept. 1995.
7. [Dasg2000] D. Dasgupta, G. Hernandez and F. Nino, 'An Evolutionary Algorithm for Fractal Coding of Binary Image', In IEEE Transaction on Evolutionary Computation, Vol 4, No. 2, July 2000..
8. [Bans93] Barnsley, M., Hurd, L., Fractal Image Compression, AK Peters, Wellesley, 1993.
9. [Hutch81] John E. Hutchinson, Fractals and Self Similarity. Indiana University Mathamatics Journal, Vol. 35, No. 5. 1981.
10. [Saup94] D. Saupe, A Guided Tour of the Fractal Image Compression Literature, Technical Report, Institut f¨ur Informatik, Universit at Freiburg, 1994.
11. [Venc94] L. Vences and I. Rudomin. Fractal compression of single images and image sequences using genetic algorithms. Manuscript, Institute of Technology, University of Monterrey, 1994.
12. [Shon91] R. Shonkwiler, F. Mendivil, A. Deliu, Genetic Algorithms for the 1-D Fractal Inverse Problem, Proceedings of the Fourth International Conference on Genetic Algorithms, San Diego (1991).

نحو تخفيض وقت ضغط الصور "فراكتال" باستعمال تقنيات الجينات الصناعية

عبد القادر قعفور و محمد كمال فرعون و عبد اللطيف رحمون ❖

كلية الهندسة – قسم حاسب آلي جامعة سيدي بلعباس، الجزائر
مخبر الهندسة التطورية و نظم المعلومات الموزعة، جامعة سيدي بلعباس، الجزائر
❖ كلية الحاسب الآلي وتقنية المعلومات – جامعة الملك فيصل
الأحساء - المملكة العربية السعودية

الملخص :

إن أدوات الجينات الصناعية و الهندسة التطورية كانت من الموضوعات الأكثر إثارة لدى الباحثين في العشر سنوات الأخيرة، حيث أنها برهنت عن تفوقها و فعاليتها في مسائل و تطبيقات البحث عن الحل الأمثل في فضاءات متعددة الأبعاد.

لقد شدت تقنيات ضغط الصور "فراكتال" أيضا انتباه الباحثين حيث شكلت تطبيقات التكنولوجيات الحديثة المعتمدة على الويب تحديا جديدا. لقد تم اقتراح عدد من تقنيات ضغط الصور من طرف الباحثين الا أنها كانت بأغلبيتها تهتم بالتوازن بين نسبة الضغط و نوعية الصورة بعد الاسترجاع.

بهذا البحث نقترح خوارزميات جينية ديناميكية لتحسين عملية ضغط صور "فراكتال" بالتركيز على نظم الدوال التكرارية (IFS). يتم استعمال ترميز خاص لكل معطيات و متغيرات الصورة في سلسلة كروموزومية محددة. كود و معطيات الخوارزميات الجينية. لقد قمنا باستعمال طريقة تقسيم الصورة بغض النظر عن حجم كائنات الصورة و موقعها في الإطار. تمكن هذه الطريقة من تسريع مرحلة الترميز بتغيير مدة الترميز أثناء البحث عن أكبر تشابهات بين كائنات الصورة.

لقد أجرينا عدد من الاختبارات لهذا البرنامج على صور "فراكتال" حيث برهنت عن تخفيض نسبي لوقت التشغيل (الضغط و فك الضغط) دون خسارة جودة الصورة و هذا بسبب البحث المتوازي عن الكائنات و المجالات (blocks and ranges) التي توفرها الجينات الصناعية.

الكلمات الأساسية: ضغط صور "فراكتال"، الخوارزميات الجينية، نظم الدوال التكرارية.