Method and Architecture for Fractal Image Compression using Multiresolution Quad-Tree

[1]A. MARTINEZ-RAMIREZ, [1]A. DIAZ-SANCHEZ, [1]M. LINARES-ARANDA, [2]J. VEGA-PINEDA
[1]Instituto Nacional de Astrofísica Optica y Electrónica
Luis Enrique Erro No.1 Tonantzintla, Puebla, C.P. 72840, México
Tel., Fax +52 (222) 2 47 05 17
[2]Instituto Tecnológico de Chihuahua
Ave. Tecnológico 2909, Chihuahua, Chih., C.P. 31310, México

Abstract: - Nowadays fractal image compression is realized by using several methods, which are mainly focused to reduce the required number of operations. The present work introduces a simple and fast multi-resolution fractal image coding method, which uses a blocks classification scheme according with their size and contrast. An novel architecture which uses that new method is also introduced. The proposed architecture take advantage of method simplicity to obtain a modular fashion. Because all the range blocks are equal-sized through multi-resolution image pyramid structure, only a kind fractal coding module achieves the whole quad-tree fractal encoding.

Key-Words: - Fractal, image coding, image compression, blocks classification, local search, similitude metric.

1 Introduction

A major problem in fractal image compression is the large number of operations involved in the image coding. To reduce the number of operations, some techniques have been proposed in previously reported works [1-5]. The compression method proposed in the present work combines several techniques to accelerate the fractal coding, and it is mainly focused to implement an simple architecture to realize fast fractal image compression and video coding.

1.1 Fractal Encoding Operations Reduction

In order to reduce the number of operations involved in fractal codification, it has been commonly used the local search for similarity between range and domain blocks, with a $(2L+1)\times(2L+1)$ window of domain blocks, centered on the range block to be codified, where $0\leq L<(N_R/2)$, and N_R is the number of range blocks in the image [1]. Despite the image quality diminishes, it can be still acceptable if a $L \geq 3$ is used.

One of the fractal parameters involved is the contrast adjustment, which is denoted by *s*. For reduced quality image applications, a fixed value can assigned to s parameter [1]. Nevertheless the image quality diminishes nearly 1 dB, in the present work an s value of 0.5 is assumed in order to simplify the architecture, because no multipliers are needed when s is required.

Some reported codification methods use block similarity search in images with low resolution, re-calculating the fractal parameters bright and contrast in the original image resolution [3-5]. Simulations show that low contrast blocks do have a well matching when they are coded in the low levels of image resolution, and do not need re-calculate their fractal parameters, as is shown in Fig. 1.

The quad-tree partition is used in fractal image to increase the compression rate. Large range blocks of the quad-tree partition are codified using contracted domain blocks of a fixed size, where both involve a large number of pixels. However, that requires a high number of pixel operations. In order to reduce the number of pixel operations, the quad-tree partition method can be used through different resolution levels in a pyramidal structure, obtained from a reduction of the original image resolution. Each partition level is assigned to a resolution level of the pyramidal structure [4-5]. The method and the architecture here described use a fixed block size for each image resolution level, allowing a fixed-size processing module to realize fractal codification along the whole quad-tree in the image pyramidal structure.

2 Blocks Classification

There are methods that realize classification of range blocks or/and domain blocks to reduce the number of comparison operations between blocks. Nevertheless, those methods need additional elements of preprocessing, and the fractal parameters obtained in the low levels of resolution must be re-calculated [3]. The method proposed in the present work performs a range blocks classification according their contrast, such that largest range blocks with lower contrasts are processed in the coarsest resolution of the pyramidal structure. The medium size range blocks with intermediate contrasts are processed in some of the intermediate resolutions, while the smallest range blocks exhibiting highest contrast are processed in the highest resolution. In that way, fractal parameters (bright and contrast) do not need be recalculated again.

The method in this work begins assigning contrast thresholds on every partition level of the quad-tree, for every resolution level at the pyramidal structure. It is assumed that an image I is codified using a quad-tree partition with N+1 levels. In the first level of quad-tree partition, codified image blocks have a size of $p2^N \times p2^N$ pixels. Beginning with i = 1, if the range block contrast, considered in the i-th partition level, is greater than level contrast threshold U_i, then the range block is divided into four range blocks of $p2^{N-i} \times p2^{N-i}$ pixels size, until i = N. If the range block contrast is minor or equal than the threshold U_N, then the block is codified into the N-level of resolution. If not, every range block is split into four blocks of p×p pixels, where each of those blocks belongs to the N+1 level in the quad-tree partition, which is the level with smaller blocks. In that level, all the regions that were not codified in the previous levels will be now codified. When a zone has been selected to be codified in the i-th level of the quad-tree, any of the blocks that this zone covers in the partition level i+1 and next levels cannot be codified again. Contrast threshold value of every partition level belong to interval [0,255]. In the present work, contrast thresholds of U_i = 40, 80, 160, for levels i = 1,2,3, where N = 3, are proposed. Thresholds can be changed to fit a given compression rate. Obtained data set shows that lower thresholds produce a better image quality, but reduces the compression rate.

3 Multiresolution Fractal Encoding

After the range blocks classification, a fractal code is obtained for each resolution level. The selected codes on subimages I_1, … , I_N, and the original image I_{N+1}, will integrate the fractal code for the whole image I. The fractal code obtained on I_1 allows approximate blocks which displays small variations when they are displayed at the resolution of the original image I. The fractal code, which is obtained on the images I_2 and I_3 allows to approximate blocks which displays medium variations when them are displayed at same resolution of image I. The fractal codes obtained on the original image approximate the blocks having a high variation. Fig. 2 shows the codified images of Lena and Cameraman using the described method on a four-levels image pyramid structure with U_i = 40, 80, 160, for levels i = 1, 2, 3.

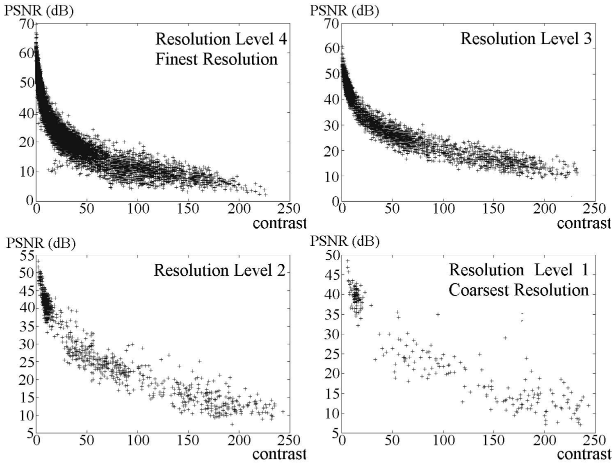

Fig. 1. PSNR vs. range block contrast for cameraman

Despite other multiresolution fractal methods have been reported in literature but, the method here introduced do not codify the image in wavelet frequency domain and do not realizes preprocessing previous to fractal coding as in [4], and the block classification scheme is simplest than scheme proposed in [3]. The advantages of method proposed in this work are: the compression rate is calculated before to the fractal coding; only a same kind of processing module achieves the fractal encoding in the whole images pyramidal structure.

4 Fractal Image Coder Architecture

4.1 Organizing the architecture

The original image is divided in M×M sectors, with 32×32 pixels each. Every sector is stored in a register bank and those sectors are subdivided in K^2 sub-sectors. Since K^2 = 4 is used in this work, every sector contains (16)×(16) range blocks, and every sub-sector contains (16/K)×(16/K) range blocks. Range blocks have 2×2 pixels, while domain blocks have 2×4 pixels, respectively. Every single sector is processed by one processing unit, which has K^2 fractal codification modules, where each module codifies a sub-sector of the image. All the fractal code sets generated by every processing unit form the whole image fractal code.

4.2 The Processing Unit

A block diagram of a single processor unit is shown in Fig. 3. Each processing unit has four fractal codification modules, which perform in a parallel way, the codification of four range blocks, of the assigned image sub-sectors. An address decoder

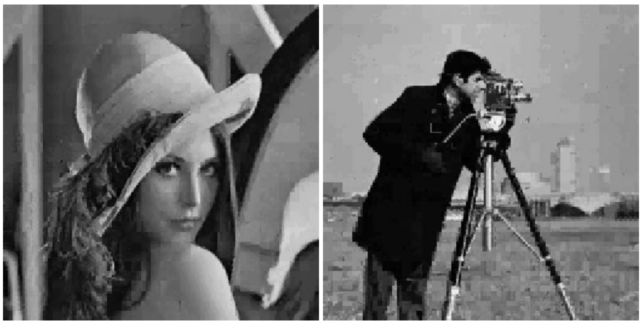

Fig. 2. Lena (PSNR = 27.6 dB, 0.21 bpp) and Cameraman (PSNR = 24.37 dB, 0.21 bpp) using proposed method.

selects every pixel to extract a range block, which comes from a single sub-sector of the image, or a domain block, which is integrated by pixels of four sub-sectors. The address of every pixel in the registers bank of the processing unit is composed of two parts: the first specifies the sub-sector where the chosen pixel belongs, and the other part specifies the relative position of the pixel in the image sub-sector.

4.3 The Fractal Coding Module

The fractal coding module is the basic unit of the presented architecture, and it realizes the basic operations set in fractal codification The proposed architecture for fractal coding module is shown in Fig. 4.

Every fractal coding module receives from the processing unit a range block, where each of them has a different range from the other modules. Every set of four fractal coding modules receives, from the processing unit, one domain block, which is rotated or reflected by the coding module. The luminance transformation for domain blocks is realized without multipliers, since a fixed value was chosen for the contrast adjustment parameter. To store the pixels of a domain block, eight registers are used, as opposed to other methods. Since chosen method uses chessboard pixels down-sampling, domain block has only pixels whose sum of the row and the column in which the pixel is located, results an odd number. That down-sampling is used to obtain a better image quality than the obtained using decimated domain methods. On the other hand, it is faster than no-contracted domain methods, so obtaining an image of intermediate quality. The chessboard-like downsampling improves the matching between ranges and transformed domain blocks using the metric mean absolute difference (MAD). In order to calculate the MAD, each pixel from a range block is compared with two pixels from the corresponding domain block, and its absolute differences is calculated and added, to compute the mean absolute deviation existing between those image blocks. The use of two adders allows the pipelined computation of the difference between pixels from a range block and from a domain block. The accumulation of this difference to compute the MAD of every range block, reduce almost to half the time required to realize this task. The first adder is part of an two-input accumulator loop. The second adder is part of a block which specific task is to calculate the mean absolute deviation for every domain block. The module of fractal codification also have a comparator module, which realizes the comparisons when mean absolute deviation is calculated, selects the best transformations and realizes the blocks classification.

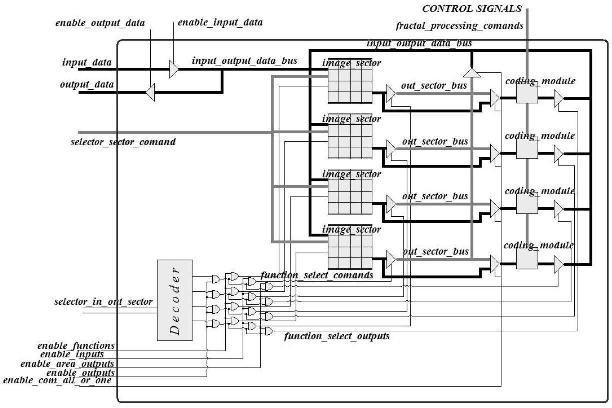

Fig. 3. Processing unit architecture.

5 Results

Table 1 shows the operations performed by one fractal coding module and the clock cycles required to perform each operation. To compare every range block to its corresponding domain block, along his eight isometric transformations, the module need 90 clock cycles. Since fractal codification module compares 8×8 range blocks against 15×15 domains blocks, the codification of an image sub-sector requires 1.296e+6 clock cycles. Table 2 shows the relation between clock cycles required to codify a single quad tree partition level in function of the search window size, for different degrees of parallelism, which are indicated for the number of fractal coding modules (FCM) inside the coding unit (CU). To codify three levels of resolution are needed 3.888e+6 clock cycles. With a clock frequency of 160 MHz, the codification is realized in 24.3 ms, enough to consider the coding of video frame sequences for real time applications. Table 3 shows the clock frequency required to process video frames in real time, as a function of the used method and the searching window size. The third column illustrates the case of the complete architecture, when four

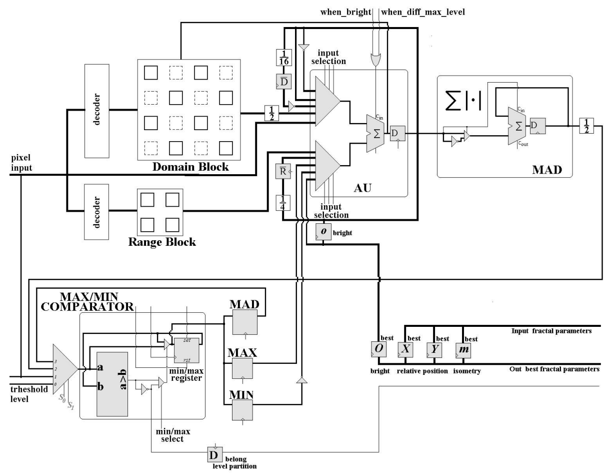

Fig. 4. Architecture of fractal coding module

fractal codification modules, two adders and down-sampling in chessboard are used for domain blocks. The results are presented for different windows of local search for similarity. In the present work L=7 has been selected. The presented architecture has been simulated using Active VHDL and iSE4.2 HDL. A single processing unit architecture was synthesized for its implementation on a VIRTEX FPGA XSV800 card, where the number of gates needed to fulfill the implementation is 39527. The implementation in FPGA for one fractal coding module uses 3986 gates.

Table 1. Operations performed by fractal coding module and the clock cycles required to realize that.

Fractal coding module operations	Clock cycles
Range capture and averaging	4
Domain capture and averaging	8
Bright calculate $o = R - sD$	1
Luminance transformation $D' = sD+o$	8
Range spatial transformation (isometries) and MAD calculation	8×8
MAD comparison	1
Fractal parameters transfer	4
Total	90

Table 2. Clock cycles vs. search windows size, for several grades of parallelism.

	Searching window size							
	L= 0	L = 1	L = 2	L = 3	L = 4	L = 5	L = 6	L = 7
FCM/CU	Clock cycles to codify a single quad tree level							
	×10	×10	$\times10^3$	×10	×10	×10	×10	$\times10^3$
256	9	81	2.25	441	729	1089	1521	20.25
64	36	324	9	1764	2916	4356	6084	81
16	144	1296	36	7056	11664	17424	24336	324
4	576	5184	144	28224	46656	69696	97344	1296
1	2304	20736	576	112896	186624	278784	389376	5184

Table 3. Clock frequency needed to obtain real time processing for video frame sequences.

	L = 0	L = 1	L = 2	L = 3	L = 4	L = 5	L = 6	L = 7
clock frequency MHz	0.47	4.7	13	25	44	65	91	119

6 Conclusions

A simple and fast method and architecture for fractal codification has been presented, which uses quad-tree partition on multiple levels of image resolution. The proposed architecture does not need additional preprocessing modules to classify the ranges blocks, and take advantage of method's simplicity. Implemented in a modular fashion, it uses a single kind of processing module to perform all the processing tasks. That characteristic allows its implementation in FPGA cards or in a custom integrated VLSI circuit. Furthermore, using an adequate clock selection, the proposed architecture can be used for real time video processing. The architecture can be configured to obtain several compression rates, by further modification of the contrast thresholds adjustment, as opposed to others reported architectures which obtain fixed compression rate [6].

References:

[1] Ning Lu. *Fractal Imaging* (Cap. 6, pp. 152 – 159, Academic Press, 1997).

[2] M. Gharavi-R. De Nardo-Y. Tenda-T. Huang, "Resolution Enhancement of images using fractal coding" in Visual Communications and Image Processing '97, J. Biemond and E. J. Delp, Eds. San Jose, CA, USA, Feb 1997, vol. 3024 of SPIE Proceedings, pp. 1089 – 1100.

[3] D. Saupe, "Accelerating Fractal Image Compression by Multi-Dimensional Nearest Neighbor Search", *Proceedings DCC'95 Data Compression Conference,* J. A. Storer, M. Cohn (eds.), IEEE Computer Society Press, March 1995.

[4] R. Rinaldo and G. Calvagno, "Image coding by block prediction of multiresolution subimages", *IEEE Transactions on Image Processing,* vol. 4, no. 7, pp. 909 – 920, July 1995.

[5] H. Lin, and A. N. Venetsanopoulos, "A pyramid algorithm for fast fractal image compression," *Proceedings of 1995, IEEE International Conference on Image Processing*, vol.3, pp. 596-599, Washington, USA, 1995

[6] D. Vidya, Ranjani Parthasarathy, T.C. Bina, N.G. Swaroopa, "Architecture for fractal image compression". *ELSEVIER Journal of Systems Architecture,* No. 46, pp. 1275-1291, 2000