CASCADE FRACTAL IMAGE COMPRESSION AND ITS MODIFICATION

Nikolay N. Ponomarenko[1], *Karen Egiazarian*[2], *Vladimir V. Lukin*[1], *Jaakko T. Astola*[2]

[1]Dept 504, State Aerospace University (Kharkov Aviation Institute)
17 Chkalova Street, 61070, Kharkov, UKRAINE,
Tel/fax + 38 0572 441186, E-mail lukin@xai.kharkov.ua
[2]Tampere International Center for Signal Processing, Tampere University of Technology
P.O.Box-553, FIN-33101, Tampere, FINLAND, Tel. +358 3 365 3860,
Fax +358 3 365 3857, E-mail karen@cs.tut.fi, jta@cs.tut.fi

ABSTRACT

We propose an approach to fractal image compression that provides fast decoding of the compressed image in one iteration and allows knowing the accurate value of the error contributed by each range block to the collage error at each step of partition scheme optimization. A modification of this method assuming equal sizes of domain and range blocks is considered. The results of the proposed approach application to test images are analyzed. The further research directions are discussed.

1. INTRODUCTION

Fractal image compression (FIC) is an area of intensive research during recent decades [1,2]. It is established that several severe shortcomings are inherent to FIC. One of the strictest drawbacks is that extensive computations are required at the stage of image compression. In this sense, the time consumption needed for FIC is usually considerably larger than for some other compression techniques.

Another drawback is the rather large computational expenses spent for fractal image decompression that are due to relatively large number of iterations needed for decoding. Besides, without decoding at the stage of partition scheme optimization at compression stage, it is impossible to know the error contributed due to non-accurate mapping the range block to the domain block into collage error of image decoding.

The latter two drawbacks are determined by the presence of chains and cycles in mapping. For range block decoding, it is necessary to find the values for range blocks that can be placed at the area occupied by the corresponding domain block. In turn, for their calculation it is necessary to calculate the values for range blocks located in places occupied by the corresponding domain block, etc. Fig.1 shows an example of the cycle where the range blocks R1, R2 and R3 correspond to the domain blocks D1, D2, D3, respectively. Then for getting R1 one should have R2, for obtaining R2 one should have R3, and, for getting R3 it is required to know the values of R1. And this is a rather simple case.

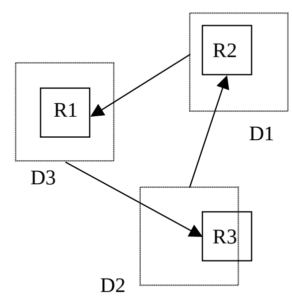

Fig.1. An example of the cycle in block mappings

The real image compression situations one runs into in practice are often much more complicated than the example in Fig.1. Usually for calculation of the values of one range block it is necessary to use in chain the pixels of many or even all image range blocks in chain including the considered range block itself. Therefore, several iterations are to be done in image decompression for getting an appropriate accuracy. This procedure is computationally inefficient and it makes practically unreal to calculate the changing of collage error for each variant of each block splitting for each step of partition scheme optimization.

An idea of the proposed cascade fractal image compression (CFIC) consists in such organization of mapping the range blocks to domain blocks that does not assume the presence of any cycles in mapping and chains inside cascade regions. This permits just at the stage of partition scheme optimization to accurately know the error E_{BD} contributed by each range block into decoding collage error, where $E_{BD} = \sum_{i=1}^{N} (\boldsymbol{B}_i\text{-}\boldsymbol{D}_i K\text{-}C)^2$, $\boldsymbol{B}_i$ is the i-th range block, $\boldsymbol{D}_i$ denotes the best match domain block, N is the number of range blocks, K denotes the contrast coefficient, C is the intensity. This also allows one to decode the compressed image in one iteration without making worse the decoding accuracy.

2. PRINCIPLES OF CASCADE FRACTAL COMPRESSION

To avoid the cycles in range block to domain block mapping an image is divided into separate lots (regions) that are altogether similar to a "waterfall cascade". In coding of every cascade region, as the domain blocks only those cascade regions are used that are already coded to the given moment.

At the very beginning of coding, a 4x4 pixel fragment placed in one of four possible image corners is selected. Fig. 2 gives an example when the coding starts from the upper left corner of an image.

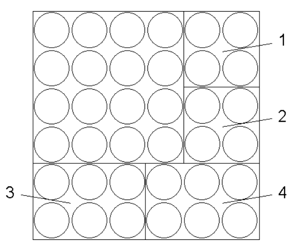

Fig. 2. An initial fragment of size 4x4 pixels and the principle of forming the first and the following (next) cascade regions.

This fragment is used as a domain block for coding the regions 1 and 2 that have the size 2x2 pixels and are attached to the initial fragment from the right side. After coding the regions 1 and 2 it becomes possible to use three regions as the domain block set – the initial one as well as the regions 1 and 2 (the total size is then 6x4 pixels). That is why the next regions 3 and 4 to be coded (the next regions are always attached to the more long side of the formed rectangular of the domain block field) have the size 3x2 pixels. This process is continued until the entire image is divided into regions. The areas of cascade regions rapidly grow. Fig.3 shows the order of compression and the speed of region area growing for an image with entire dimension 512x512 pixels. The numbers inside the cells show the order of their encoding.

For optimization of each cascade region compression, this region partition scheme is applied. An important advantage of this approach is a very high speed of image decoding. That is why to avoid decreasing of decoding speed it seems reasonable to use the partitioning schemes with only rectangular shapes of blocks - quadtree, horizontal-vertical and similar ones [2]. In this case the computational expenses for pixel search inside the range block at decoding stage will be minimal.

The schemes of cascade region partitioning are optimized sequentially according to the order of region number in cascade. Different approaches to memory resource distribution between cascade regions are possible. The simplest in implementation seems to be an indirect distributing that takes into account the limit of losses in image compression. In this case for all cascade regions the common acceptable level of losses is preset. The partitioning scheme is then split into details until the losses for a given region become smaller than the preset value. Thus, the partition scheme for more complicated regions should contain larger amount of blocks than the partition schemes for more simple regions.

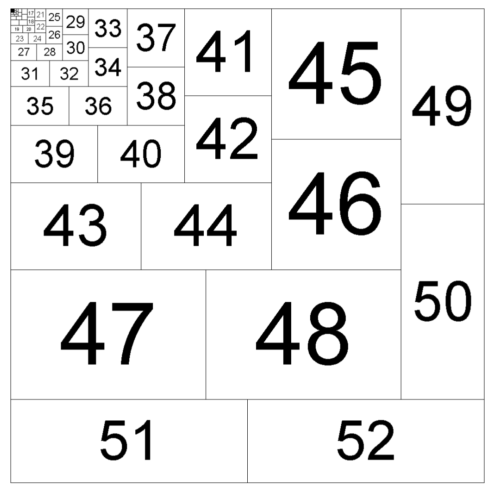

Fig. 3. An order of compression of cascade regions for an image of size 512x512 pixels.

At the stage of image decoding, the cascade regions are to be decoded in the same order as it was executed at the coding stage. That is why to the moment of the given region decoding starting the values for regions used as domain blocks are already known.

3. MODIFICATION WITH EQUAL SIZES OF DOMAIN AND RANGE BLOCKS

As it was already mentioned the cascade approach to image coding avoids the necessity in cycle presence in the scheme of domain block to range block mapping. This is why, the domain block scaling required for conventional method to provide the convergence of image iterative decoding becomes now not necessary. It becomes possible to use the range blocks and domain blocks of equal size (see Figures 4 and 5). This opportunity in case of cascade compression allows having smaller number of cascade regions and, therefore, to improve the quality of each image region coding as well as the quality of the entire image coding. This also allows us to increase the speed (to decrease the computational time) of image decoding due to the absence of domain block scaling operation.

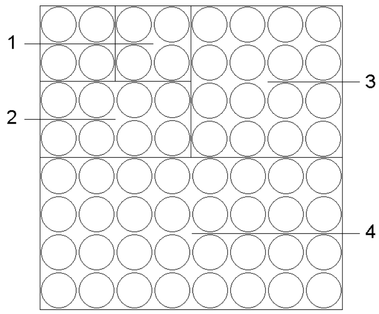

Fig. 4. An initial fragment of size 2x2 pixels and the principle of forming the first and the next Regions of Cascade.

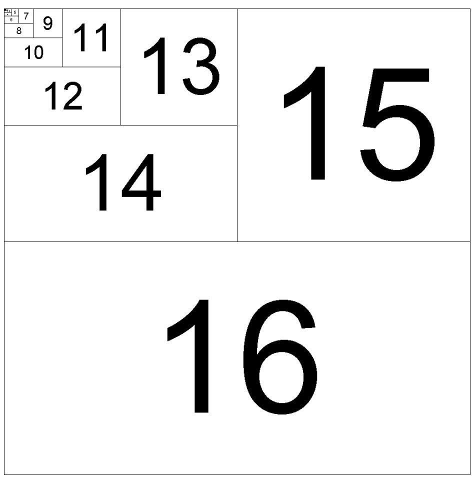

Fig. 5. An order of cascade regions compression for equal sizes of range blocks and domain blocks in an image of size 512x512 pixels.

For CFIC and its modification with equal sizes of domain and range blocks (MCFIC) the field of domain block search for given range block makes smaller. Intuitively, this should lead to decreasing coding efficiency. However, such negative effect is partly compensated due to decreasing the number of bits needed for coding the found domain block coordinates. Besides, simultaneously some increasing of coding efficiency is observed due to more accurate calculation of E_{BD}.

4. NUMERICAL SIMULATION RESULTS

Before giving the numerical simulation results let us note that for the proposed image compression method it is important in what image corner one places the first cascade fragment and, thus, the cascade starts to be formed.

We have examined all eight possible variants (including mirror transforms) and established that for image ”Lenna” the difference in compression ratio (CR) was 10% between the best and the worst obtained CR. One way out is to perform the compression for all possible variants and to select the best one. This increases the encoding time by 8 times but does not somehow results in decoding efforts. Three additional bits should be added to the encoded image file to indicate the position of the cascade start.

The numerical simulation was carried out for a set of test grayscale images и CR.

As the partition scheme, a modified variant of horizontal-vertical partition scheme was used [3]. According to this scheme, each range block could be partitioned into two equal blocks either vertically or horizontally. For each range block, the following information is stored: the domain block coordinates (up to 15 bits depending upon the codebook size), the intensity (7 bits), the contrast coefficient (5 bits) (the rotations have not been analyzed, but their use can provide additional increasing of PSNR by 0.1-0.4 dB depending upon CR). Partition scheme data are compressed by arithmetical coding (2-2.5 bits per block) and also stored in encoded image.

Besides, the variant number of the cascade forming (3 bits) was recorded for the entire image. Table 1 shows the results for image coding using the proposed cascade fractal coding technique and its modification.

Table 1.

Image	Bit rate	PSNR, dB		
		FIC	CFIC	MCFIC
Lenna	0.02	23.28	22.35	22.94
	0.1	28.60	28.19	28.64
	0.5	35.45	35.61	35.80
Barbara	0.02	21.13	20.71	21.22
	0.1	23.57	23.45	23.81
	0.5	28.39	29.68	29.90
Baboon	0.02	19.34	18.89	19.46
	0.1	20.45	20.40	20.73
	0.5	23.48	23.98	24.27
Goldhill	0.02	23.22	22.63	23.17
	0.1	27.01	26.59	26.96
	0.5	31.88	31.89	32.10
Peppers	0.02	21.96	21.49	22.04
	0.1	28.11	27.27	27.66
	0.5	34.40	34.38	34.59

As seen, for bpp=0.02 the quality of decoded images for CFIC and MCFIC can be a little bit poorer than for FIC.

This, as said earlier, is explained by smaller number of analyzed domain blocks. At the same time, for large bpp (0.1, 0.5) higher accuracy of evaluation of E_{BD} results in considerable (up to 1…1.5 dB) improvement of decoded image quality. In such cases the quality for MCFIC is always by 0.2…0.5 dB better than for CFIC.

We have also estimated the average decoding time for different bit rates (see the Table 2). The estimations have been done for computer with the processor Duron 800 MHz. The decoding program was designed using Delphi with Assembler fragments (without using MMX and other special commands able to speed up the image decoding).

Table 2.

Bit rate	Average decoding time, ms		
	FIC	CFIC	MCFIC
0.02	265	10.3	9.5
0.1	303	12.7	12.0
0.5	397	14.9	14.5

As seen, the decoding time sufficiently depends on bpp and, respectively, on the number of range blocks in an image. The decoding time for FIC is approximately 25 times larger due to using iterations at decoding stage (in the considered case 20 iterations have been used for providing desirable accuracy).

CONCLUSIONS

The investigations carried out by us and the obtained results of numerical simulations let us make conclusions concerning two basic advantages of the new proposed approach [4]. Very high decoding speed is provided simultaneously with high enough quality of image compression. This clearly shows the basic perspectives and applications of the proposed method of cascade image compression and decoding. For example, they can be effectively used in video compression where the high speed of decoding is of prime importance.

A drawback of the proposed approach is the large computation time required for image coding. However, this drawback is typical for many methods of fractal compression. There are several ways how to solve this problem. It is possible to apply some traditional methods of fractal compression speeding up, in particular, those ones described in [2], [5], [6]. One more opportunity that seems to be a good possible solution is to use hardware methods of compression speeding up. The basic computational load is the search procedure for range block to domain block mapping. But this procedure can be executed in parallel. All the field of the domain blocks (virtual codebook) can be divided into several equal areas (parts) and the search of the best mapping between the range blocks and the domain blocks can be done separately by several specialized processors in the corresponding area.

For such specialized processor one has to input the range block coordinates and its size as well as the coordinates and the size of the virtual codebook part where the search should be done. The processor output contains the following data: the coordinates of the domain block the range block should be mapped to and the mapping parameters. Four specialized processors of this type allow decreasing the coding time by four times and, in general, the time required for image encoding occurs to be approximately inversely proportional to the number of specialized processors operating in parallel.

Of course, such hardware can be very expensive. However, it is needed only for video data encoding. For decoding one needs to have only the software program that does not require hardware support for its operation acceleration. Moreover, the complexity and high cost of hardware can serve as a barrier against illegal manufacturing of video information. This is one more advantage in favor of the proposed approach.

The data presented in Table 1 confirm the high quality of compression. However, there are many ways to further improve it at both stages of image encoding and decoding. They are, in particular, the following: to apply more complicated partition schemes (possible improvement is about 0.1-0.4 dB [3]) and better algorithms of their optimization, to use more sophisticated algorithms of memory resource distribution between the cascade regions, to smooth artifacts occurring at range block edges [7], etc.

REFERENCES

[1] *Fractal Image Encoding and Analysis*, Edited by Yuval Fisher, Berlin Heidelberg, Germany: Springer Verlag, 1998.

[2] B. Wohlberg, G. Jager, "A Review of Fractal Coding Literature", *IEEE Trans. On Image Processing*, Vol. 8, pp. 1716-1729, Dec. 1999.

[3] N. Ponomarenko, V. Lukin, K. Egiazarian, J. Astola, "Modified horizontal vertical partition scheme for fractal image cpmpression", *CD-ROM Proc. of 5-th Nordic Signal Processing Symposium*, on board Hurtigruten, Norway, Oct. 2002, 4 pages.

[4] Patent № FI 113603 B Fractal compression method / N. Ponomarenko, K. Egiazarian, V. Lukin, A. Zelensky, J. Astola // Patenttilehti, №9, p. 33, 14.05.2004.

[5] D. Saupe, H. Hartenstein, "Lossless Acceleration of Fractal Image Compression by Fast Convolution", *Proceedings of IEEE*

International Conference on Image Processing, Vol. 1, pp. 185-188, Sep. 1996.

[6] N. Ponomarenko, K. Egiazarian, V. Lukin, "Acceleration of Fractal Image Compression by Correlation Trees", *Proc. of the First South-East European Symposium on Interdisciplinary Approaches in Fractal Analysis*, May 7-10, 2003, Bucharest, Romania, pp.67-72.

[7] K. Egiazarian, M. Helsingius, P. Kuosmanen, J. Astola, "Removal of blocking and ringing artifacts using transform domain denoising", *Proc. of ISCAS'99,* vol. 4, May 30 - June 2 , 1999, pp. 139 – 142.