SPEEDING-UP THE FRACTAL COMPRESSION WITH CLUSTERING

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@mmds.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

A problem of speeding-up the fractal compression of images is discussed. The techniques based on clustering are considered. An approach to clustering based on median split method is proposed. A procedure of reassigning the domain blocks between clusters is shown to improve the PSNR or to increase the compression ratio. The optimal number of clusters that minimizes the computation efforts for this procedure is derived. Some practical limitations of clustering are discussed and the particular ways out are given.

1. INTRODUCTION

Fractal image compression is an intensively developing topic in modern image processing [1]. Along with wavelet compression [2] it has drawn attention of many researchers and exhibited some advantages as a challenge to already used standard compression techniques.

The problems in fractal compression of still images are a very large amount of computations (and time) required and, thus, a necessity of speeding up the compression process without essential loss in quality [3], [4] as well as the recovered image enhancement. This paper mainly addresses the first problem dealing with speeding up the search of optimal domain blocks for range blocks by means of clustering.

There exist a lot of different strategies of domain search [3], but the strategy based on clustering appears to be the most reasonable. First, due to its application it becomes possible to obtain a rather great benefit in compression speeding up with reasonably small degradation of recovered image quality (or, if desirable, with small reduction of compression ratio). Second, this technique is easily adjusted to other methods of compression speeding up or image quality enhancement.

So below we first consider an approach described in [4] more in detail since our proposed methods are intended to improve it. Then an application of median split algorithm to data clustering is considered and its basic steps are given. Its advantages and drawbacks are analyzed. The procedure of reassigning the domain blocks between clusters is put forward and shown to be rather effective for PSNR increasing. All the benefits and drawbacks are illustrated using numerical simulation results for a set of traditional test images.

2. BASIC STEPS IN CLUSTERING

In standard fractal compression a dominant part of time (almost 100%) is spent on finding the correspondence between the compressed range blocks and the domain blocks [1]. Denoting a number of range blocks as K and a number of domain blocks as N the complexity of compression becomes of $O(K*N)$ since it is required to compare each range block with each domain block. The use of clustering permits to avoid a linear search between all variants due to comparison of each range block with only the cluster centers and, then, with domain blocks of the cluster being the closest to the considered range blocks.

To perform these operations Wein and Blake [4] propose the following steps to be done in fractal compression: 1) the calculation of the number of clusters optimal for given number of domain and range blocks, 2) the division of the virtual codebooks (a domain block set) on clusters using the selected clustering method; 3) the search of the nearest cluster (according to its center) for each range block; 4) the search of the optimal domain block for each range block within the determined cluster.

The application of this scheme permits to decrease the complexity of domain search if the dimensions of virtual codebook clusters are equal. If initially it was $O(K*N)$, then, it becomes of order $O(K*M+K*N/M)+Ocl$ where M is the number of clusters and Ocl describes the complexity of clustering algorithm. As shown in [4], the value of the sum $K*M+K*N/M$ is minimized in case of $M=N^{0.5}$. The complexity is expressed then as $O(2*K*N^{0.5})+Ocl$.

Taking this into account one gets the following requirements to clustering algorithm:

1) it should provide the predetermined number of clusters desirably of equal size;

2) the complexity of clustering algorithm has to be considerably less than $O(K*N-K*M-N*M)$ in order to ensure considerable benefit in speeding up the fractal compression in comparison to methods based on linear (full) search;

3) the algorithms are to be able to operate with vectors having large dimensions (like 256 or 1024).

The first and the second requirements can be satisfied for several algorithms. Unfortunately, among these algorithms practically no one is able to satisfy the last condition. This results in noticeable degradation of recovered image quality in comparison to compression based on linear search. In particular, in [4] it is proposed to apply the *pnn*-algorithm with complexity $O(N*logN)$. However, to our opinion, the application of other methods could be more expedient.

3. MEDIAN SPLIT CLUSTERING

The median split clustering algorithm contains the following steps:

1) the calculation of the value variance for each vector coordinate (component);

2) the calculation of the median for the vector coordinate having the largest variance;

3) the division of domain field into two clusters having almost equal numbers of domains (the domains with the value for the considered coordinate less than median are referred to one cluster and otherwise);

4) until the number of clusters reaches the required value the steps 1-3 are repeated.

If the number of clusters is not the power of 2 the obtained clusters do not have equal sizes. However, this drawback is compensated by the higher speed of the considered algorithm. Actually, the median split algorithm complexity is $O(N*logM)$. That is why, if, for instance, the number of donain blocks is 65536 and the number of clusters is 256 the median split algorithm of clustering is approximately two times faster than *pnn*-algorithm.

The abovementioned technique of speeding up the fractal compression permits to get a very good (about $0.5N^{0.5}$) benefit in time needed. One more advantage is that the benefit almost does not depend on the range block number, but only upon the domain block number. Unfortunately, a considerable loss in recovered image quality (characterized by PSNR) is observed.

Let us confirm this by numerical simulation results. Table 1 presents the results of fractal compression for the test image (512x512) Lenna using the linear search method and two schemes of clustering – without and with reassigning the domain blocks (See next Section). The compression method performance is characterized by the values of PSNR for the same compression ratio (CR). It is seen (columns 3 and 4 of Table 1) that with the cluster number increasing the loss in PSNR becomes larger and for optimal value of M (256) the loss is of about 0.7 dB and it can be even larger. We have checked this for other traditional test images like "Mandrill" and "Peppers" and observed both the same tendency and the same order losses in PSNR.

One of the reasons is the necessity to perform the vector quantization for vectors with large dimensions. For median split algorithm the division of the vector field is done using only few vector coordinates (components) while the other ones remain unanalyzed. This leads to considerable inhomogeneity of the cluster content and the high probability of false decisions in clustering and comparison making. When the number of clusters increases and computations reduce, the PSNR decreases since the range blocks and the domain blocks the most close to them could occasionally drop into different clusters.

4. DOMAIN BLOCK REASSIGNING

Let us consider what can be done to get around this problem or, at least, to minimize its consequences. First, let us take into account the fact that for range blocks we find the closest cluster center and for domain blocks this procedure is not executed. They are considered to belong to that cluster to which they have been referred after initial clustering. But in practice a large percentage of domain blocks occur to be placed more close to the centers of another, neighbor clusters. So if after the initial clustering one makes for each domain the search of the closest cluster center the quality of finding the correspondence between the range blocks and the domain blocks can increase (although the complexity of search then increases by $O(N*M)$.

In this case the scheme of clustering for fractal compression is the following:

1) the calculation of cluster number optimal for given numbers of domain and range blocks;

2) the division of virtual codebooks (the domain block set) into clusters using the selected clustering algorithm;

3) the search of the nearest cluster (according to the distance to its center) for each range block;

4) the search of the nearest cluster (according to the distance to its center) for each domain block and translation of the domain block to this cluster if needed;

5) the search of the optimal domain block for each range block within the determined cluster.

The application of this scheme results in considerable decreasing of loss in the recovered image quality caused by clustering. Table 1 presents the corresponding results (see columns 5 and 6). The range blocks for all the considered methods have the size 8x8, the domain block number is 62001, the clustering has been performed by median split algorithm.

It is seen from this Table that the increasing of the cluster number for the proposed clustering method with reassigning also results in reduction of PSNR in comparison to linear search. However, if the domain block reassigning is not used the loss in PSNR is about 0.3 dB larger than if the reassigning is applied (compare the results in columns 3 and 5 or analyze data in column 7). Moreover, the number (percentage) of domain blocks that have changed the cluster is large – from 60 to 72%.

Note that because of changing the clustering complexity – for clustering with domain block reassigning it became $O(K*M+K*N/M+K*N/M)+ Ocl$ – the optimal number of clusters is not equal to $N^{0.5}$. It depends on both the domain and range block numbers. In situations when $N>>K$ it is reasonable to select $M=K^{0.5}$ and if N and K are of the same order one should use $M=N^{0.5}$. The latter situation takes place in fractal compression using adaptive partition schemes [3] where the domain block number is restricted to get an opportunity of image partitioning on larger number of range blocks and its optimization. However, for majority of practical tasks one has $N>>K$ and the search complexity can be approximately expressed as $O(K*K^{0.5}+2*N*K^{0.5})+Ocl$.

Therefore, the procedure of domain block reassigning permits to partly solve the problem of recovered image enhancement. However, the dimensions of clusters become not equal and this can lead to some growth of computations to be performed. So one of problems to be considered in future is the application of methods providing better clustering, for instance, due to more perfect vector quatization approaches.

5. OTHER POSSIBLE SOLUTIONS

One heuristical way out is proposed below. Suppose we do not subtract the mean for every domain block before finding the median value and cluster splitting into two parts. Then the domains with similar (approaximately equal) coordinate values will fall down into the same cluster. If one deals with mosaic image compression then it is reasonable to assume the following. If the means for different domain blocks are rather close then these domain blocks are with larger probability placed in image not far away from each other than the domains with considerably different means. In turn, it means that the other vector coordinates of these domains can be highly correlated. If these assumptions are valid then the clustering by one coordinate at every step is "supported" by clustering for some other dependent coordinates of the vector.

Let us analyze the obtained numerical simulation results. Some of them are given in Table 2.The time required for finding the range and domain block correspondednce for equal size clusters is considered as T. The ratios of computation time in case of clustering with subtraction of mean and without subtraction are denoted as T_s/T and T_w/T, respectively. The number of domain blocks is given in the first column, the PSNR values are computed for range block size 8x8 and 64 clusters.

The test image is the same – Lenna 512x512 pixels. It is worth noting here that all the numerical simulation results in this paper are obtained for the following technique of color-to-gray scale image transformation $Gr=0.3R+0.59G+0.11B$ where R, G and B are the intensities of red, green and blue components of color image, respectively. We would like to mention also that the results (the PSNR values) can be noticeably (even by 1-2 dB) different for other variants of color-to-gray scale transformation.

As seen from Table 2 there is almost no difference in PSNR values for the considered clustering methods. The increase in computations with respect to ideal case is not too large, however, in the latter case (when the mean values are not subtracted) the increase is smaller due to better clustering with almost equal cluster dimensions.

Finally, let us give quantitative data concerning computational load for the linear search and the proposed clustering technique. Table 3 presents the results for three images - Lenna, Mandrill and Peppers. The PSNR values are given for the range block size 8x8, domain block size 16x16. The number of domain blocks is 62001. The optimal number of clusters for the considered case is equal to 62, but for suitability of the clustering algorithm operation it was selected to be 64. Three values of CR have been considered – 5.3, 21.3, and 85.3. We do not present here the time required for clustering expressed in seconds since the CPU depends upon many factors like computer type and rate, etc. That is why the ratio T_{ls}/T_c is given where T_{ls} and T_c are the time interval durations for making up the linear (full) search and search with clustering.

As seen from Table 3, far all the considered cases the difference in PSNR for compression based on linear search or clustering is not large. It does not extend 0.4 dB for small CR and reduces for larger CR values. Of course, the PSNR also depends upon image type and it is the worst (in case of equal CR) for the test image "Mandrill" which contains many textured regions. The most impressive result is the considerable speeding up the computations due to clustering. The benefit is especially large for CR values within the range of 5…20, where the computation time can be decreased by 30…60 times. Just this range of CR values is typical for many practical applications. For larger CR the ratio T_{ls}/T_c decreases but still remains large enough.

CONCLUSIONS

It is shown that due to clustering the computational complexity of the fractal compression can be significantly decreased. The median split method proposed for performing the clustering has been demonstrated to provide an appropriately small loss in PSNR with respect to the linear search fractal compression techniques for wide range of CR and different test images.

REFERENCES

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

[2] *Wavelet Image and Video Compression*, Edited by Pankaj N. Topiwala, Boston, USA: Kluwer Academic Publishers, 1998.

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

[4] C. J. Wein, I.F. Blake, "On the Performance of Fractal Compression with Clustering", *IEEE Trans. On Image Processing*, Vol. 5, pp. 522-526, March 1996.

Table 1. Comparison of performance of different compression methods for test image Lenna, CR=21.3.

Cluster number	PSNR for linear search dB	PSNR for Clustering without Reassigning, dB	PSNR loss because of clustering, dB	PSNR for clustering with reassigning,dB	PSNR loss because of clustering, dB	Loss reduction due to reassigning, dB	Precentage of reassigned domains blocks
32	30.28	29.81	0.47	30.12	0.16	0.31	61.1%
64	30.28	29.75	0.53	30.04	0.24	0.29	65.7%
128	30.28	29.67	0.61	30.01	0.27	0.34	69.7%
256	30.28	29.58	0.7	29.9	0.38	0.32	72.7%

Table 2. Comparison of PSNR and computation time for two considered techniques of clustering.

Domain block number	PSNR, dB (Mean is not subtracted)	T_w/T	PSNR, dB (mean is subtracted)	T_s/T
62001	30.04	1.154	30.06	1.091
50000	30.00	1.126	29.98	1.053
38000	29.93	1.132	29.92	1.030
26000	29.80	1.097	29.80	1.045
20000	29.71	1.084	29.70	1.025
14000	29.57	1.101	29.54	1.006

Table 3. Comparison of PSNR and computation time for compression techniques based on linear search and clustering for different test images.

Test image	CR	PSNR, dB, linear search	Number of clusters	PSNR, dB, with clustering	Difference in PSNR, dB	T_l/T_c
Lenna	5.3	36.34	128	35.95	-0.4	58.5
	21.3	30.28	64	30.06	-0.22	28.4
	85.3	26.00	32	25.80	-0.2	13.8
Mandrill	5.3	26.72	128	26.33	-0.4	57.6
	21.3	21.78	64	21.65	-0.13	28.0
	85.3	19.64	32	19.79	+0.15	14.9
Peppers	5.3	35.54	128	35.30	-0.24	59.0
	21.3	30.44	64	30.22	-0.22	29.4
	85.3	25.53	32	25.37	-0.16	14.8