ICSP2006 Proceedings

A Fast Fractal Image Coding Algorithm Based on FGSE

Meiqin Liu, Yao Zhao, Haozhuang Yang, Zhiyuan Zhang
Institute of Information Science, Beijing Jiaotong University, Beijing 100044, P. R. China
E-mail: liumeiqin_2004@126.com

Abstract

In fractal image coding, the main consuming time is the process of finding the best matched domain block in a large domain pool for a given range block. In this paper, we propose a fast algorithm for fractal image coding based on the following lemma and fine granularity successive elimination (FGSE). The lemma describes the relationship between fractal image coding and FGSE. This algorithm can decrease about half domain pool before range-domain matching computation and further improve the speed by combining with other fast algorithms of fractal image coding. Simulation results show that our algorithm takes less coding time, has absolutely the same fidelity measured by PSNR as that of Saupe's and achieves super quality compared with Fisher's algorithm.

1. Introduction

Since the fractal theory was introduced into the field of image coding by Barnsley in 1988 [1], fractal image coding was noticeable in the field of image compression due to its advantages of resolution independence, fast decompression and high compression ratio. In 1989, the first automatic fractal image compression scheme was presented by A.E.Jacquin [2]. The scheme is based on the representation of an image by a set of iterated contractive transformations for which the reconstructed image closed to the original image is an approximate fixed point.

But it suffers from a long coding time and limits its practical application. For example, for an $N \times N$ image, the number of arbitrarily sized square sub-regions is of order $O(N^3)$ and the exhaustive search for finding the optimal mappings is of order $O(N^4)$. In order to overcome this drawback, many speed-up algorithms were proposed in past twenty years. These algorithms were mainly classified into three directions: reducing the size of domain pool [2][3][4] (the range block is only compared with the domain blocks in the same class classified by the feature of the block such as mean and variance [3]), eliminating the domain blocks unmatched with the given range block based on the variance of the blocks or error function inequality [5][6][7][8] , and improving the search speed by exploiting the kd-tree data structure [9][10][11] . However, most of these algorithms can significantly reduce the required computation only at the expense of more or less degradation in the reconstructed image quality.

Recently, Zhu presented a fast optimal block matching in motion estimation, named as FGSE [13]. It is featured by providing a sequence of fine grained boundary levels in an aim to reject a checking candidate as early as possible, thus it can reduce the computational complexity. However, FGSE can not be directly used in the fractal image coding, because FGSE is determinant by the sum of absolute difference (SAD) while fractal image coding is estimated by the least squares error of a range and domain block approximation $E(R,D)$. The relationship between $E(R,D)$ and SAD will be introduced in Section 3. Furthermore, in fractal image coding, the size of range is different from that of domain blocks and the absolute value of contrast scaling s must be less than 1. However, in motion estimation， these conditions don't need to be satisfied. And Simulation results described in Section 4 show that the quality of the proposed algorithm is the same as that of Saupe's.

The rest of this paper is organized as follows. In Section 2, the basic fractal image coding scheme and FGSE are briefly reviewed. Our algorithm is given in Section 3. Simulation results of the new algorithm are described in Section 4. Finally, we summarize this paper in the concluding Section 5.

2. Review of fractal image coding and FGSE

2.1 The basic fractal image coding scheme

Nowadays, almost all of the improved fractal image coding algorithms are derived from the basic fractal image coding scheme. So above all, we will introduce the scheme and the procedure of it is shown in Fig. 1 [12].

In the scheme, firstly, the original image is divided into range blocks and domain blocks. Range blocks are non-overlapped and the collage of them could cover the whole image. Domain blocks could be overlapped and the size of them is larger than that of range blocks.

0-7803-9737-1/06/$20.00 ©2006 IEEE

Then, the best matching domain block is to be searched in a large domain pool Ω on the premise of the following equation for the given range block R.

$$E^2(R,D) == \min_{D\in\Omega}\min_{s,o}\left\|R-(s\cdot D+o\cdot I)\right\|^2 \qquad (1)$$

Apparently, the process of finding the best matches is time-consuming. And last, the necessary parameters are stored in order to decode.

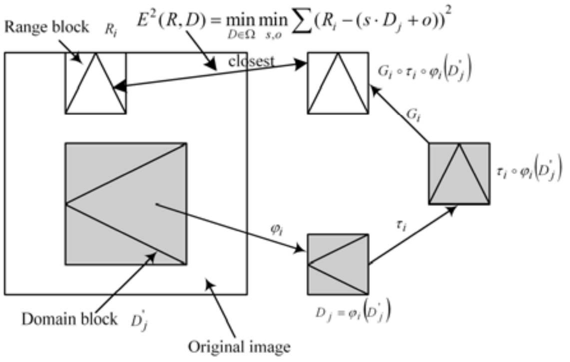

Fig.1 the process of fractal image coding

2.2 FGSE

Block matching method is widely used for motion estimation in video coding due to its simplicity and regularity. The conventional full search algorithm is the simplest block matching method, but its extremely large computation complexity makes it unpractical. Many fast block matching algorithms have been developed and FGSE is one of these algorithms. It is a fast optimal block matching in motion estimation, uses the following equation, a sequence of fine granularity scalable boundaries BV_l, to eliminate checking candidates.

$$BV_l \le BV_{l+1} \text{ and } BV_L = SAD \qquad (2)$$

where SAD, the sum of the absolute difference, is the matching error. As level l increased by 1, there is one more sub-block of size $m\times m$ being partitioned into 4 smaller sub-blocks of size $(m/2)\times(m/2)$. Equation (2) means that boundary value BV_l tends to become larger as level l increases, and all boundary values are less than or equal to the matching error SAD. And a large partial boundary values calculated at level $l-1$ can be utilized in establishing boundary values at level l. Thus it can greatly decrease the computation complexity and eliminate the unmatched blocks as early as possible. Moreover, the algorithm is lossless.

3. The proposed algorithm

As mentioned above, FGSE can not be directly introduced to fractal image coding. In this section, we introduce the lemma to describe the relationship between fractal image coding and FGSE, avoid the effect of the s on the matching error by normalizing. And thus we can use the inequality and FGSE to eliminate many unmatched domain blocks in advance.

Lemma: Let $\phi(R)$ and $\phi(D)$ be the normalized range block R and domain block D, $r_{i,j}$ and $d_{i,j}$ be the pixel intensities at location (i,j) in $\phi(R)$ and $\phi(D)$, respectively and $SAD=\sum_{i=1}^{N}\sum_{j=1}^{N}\left|r_{ij}^2-d_{ij}^2\right|$. Thus the error function is given by $E(R,D)\ge\frac{1}{2}\cdot\left\|R-\bar{R}\cdot I\right\|\cdot SAD$.

Proof: The optimal contrast scaling s and luminance shift o can be achieved by (1).

$$s=\frac{\left\langle R-\bar{R}\cdot I, D-\bar{D}\cdot I\right\rangle}{\left\|D-\bar{D}\cdot I\right\|^2}, \quad o=\bar{R}-s\cdot\bar{D} \qquad (3)$$

where $\bar{R}$ and $\bar{D}$ are the average of the pixel intensity of the range block R and the contracted domain block D.

Based on the (1) and (3), we can get

$$E^2(R,D)=\left\|R-\bar{R}\cdot I\right\|^2-s^2\cdot\left\|D-\bar{D}\cdot I\right\|^2. \qquad (4)$$

From (3), (4), we obtain

$$E^2(R,D)=\left\|R-\bar{R}\cdot I\right\|^2\left(1-\frac{<R-\bar{R}\cdot I, D-\bar{D}\cdot I>^2}{\left\|R-\bar{R}\cdot I\right\|^2\cdot\left\|D-\bar{D}\cdot I\right\|^2}\right)$$

$$=\left\|R-\bar{R}\cdot I\right\|^2\left(1-\left\langle\phi(R),\phi(D)\right\rangle^2\right)$$

$$=\left\|R-\bar{R}\cdot I\right\|^2\left(1-\left\langle\phi(R),\phi(D)\right\rangle\right)\cdot\left(1+\left\langle\phi(R),\phi(D)\right\rangle\right),$$

where $\phi(R)=\frac{R-\bar{R}\cdot I}{\left\|R-\bar{R}\cdot I\right\|}$, $\phi(D)=\frac{D-\bar{D}\cdot I}{\left\|D-\bar{D}\cdot I\right\|}$.

Since $\sum_{i,j=1}^{N\times N}r_{ij}^2=\sum_{i,j=1}^{N\times N}d_{ij}^2=1$, we have

$$E^2(R,D)=\frac{1}{4}\left\|R-\bar{R}\cdot I\right\|^2\cdot\left(\sum_{i=1}^{N}\sum_{j=1}^{N}(r_{ij}-d_{ij})^2\right)\cdot\left(\sum_{i=1}^{N}\sum_{j=1}^{N}(r_{ij}+d_{ij})^2\right)$$

By the well-known Cauchy inequality, we get

$$E^2(R,D)\ge\frac{1}{4}\left\|R-\bar{R}\cdot I\right\|^2\cdot\left(\sum_{i=1}^{N}\sum_{j=1}^{N}\left|r_{ij}-d_{ij}\right|\cdot\left|r_{ij}+d_{ij}\right|\right)^2,$$

$$E(R,D)\ge\frac{1}{2}\cdot\left\|R-\bar{R}\cdot I\right\|\cdot\sum_{i=1}^{N}\sum_{j=1}^{N}\left|r_{ij}^2-d_{ij}^2\right|,$$

and $$E(R,D)\ge\frac{1}{2}\cdot\left\|R-\bar{R}\cdot I\right\|\cdot SAD. \qquad (5)$$

The proof is completed.

In FGSE, if $N\times N=16$, the number of partition is $L=5$. And then we obtain the following inequality [13]

$$SAD = BV_5 \geq BV_4 \geq BV_3 \geq BV_2 \geq BV_1 \geq BV_0 \qquad (6)$$

where $BV_l = \sum_{i=0}^{S_l - 1} \left| a^{(k)}_{n(k,l)\times n(k,l)} - b^{(k)}_{n(k,l)\times n(k,l)} \right| \quad (l = 0,1,2,3,4,5)$, $a^{(k)}_{n(k,l)\times n(k,l)}$, $b^{(k)}_{n(k,l)\times n(k,l)}$ denote the square sum of the k^{th} sub-block at l^{th} level in $\phi(R)$ and $\phi(D)$, respectively. $n(k,l)\times n(k,l)$ is the size of the k^{th} sub-block at level l. And $n(k,l)\times n(k,l)$ may be $m\times m$ or $(m/2)\times(m/2)$ depending on sub-block index k and level l.

This implies that we can obtain a total of 5 boundaries and SAD is calculated at level 5. Fig. 2 shows the partition process of a block. And the rule of the partition is to do the partition sequentially from left to right and top to bottom subject to that smaller size of sub-blocks will not be partitioned further until all the other larger sub-blocks have been partitioned.

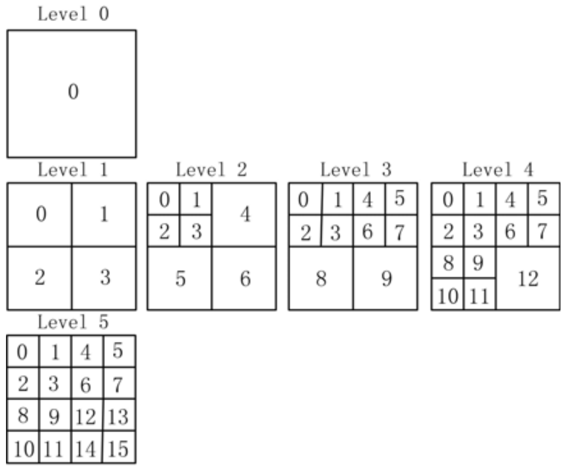

Fig. 2 partition process in FGSE

Combine (5) and (6), we get:

$$2\cdot E(R,D)/\left\| R - \bar{R}\cdot I \right\| \geq SAD = BV5 \geq BV4 \geq BV3 \geq BV2 \geq BV1 \geq BV0 \qquad (7)$$

For any range block R, $\frac{1}{2}\cdot\left\| R - \bar{R}\cdot I \right\| \geq C > 0$ is always correct. From (7), for any range block R and any domain block D, we notice that, if $E(R,D)$ is small enough, and then SAD is small enough. Whereas if they differ greatly, then $E(R,D)$ might be too large for R and D to constitute a close match. In other words, (7) implies that the range block R and the domain block D can not be closely matched unless their SAD s are as close as possible. The procedure for evaluating one range block in our algorithm is summarized as follows:

1) Let $E(R,D)_{\min} \to \infty$, $Tol = 2\cdot E(R,D)_{\min} / \left\| R - \bar{R}\cdot I \right\|$ and initialize level $l = 0$.

2) Calculate the boundary value BV_l. If $BV_l \geq Tol$, go to step 5). Otherwise, set $l = l + 1$.

3) Repeat step 2) until $l = 5$. If $BV_5 \geq Tol$, go to step 5). Otherwise, calculate $E(R,D)$.

4) If $E(R,D) \geq E(R,D)_{\min}$, go to step 5). Otherwise, update $E(R,D)_{\min}$ with $E(R,D)$.

5) Check the next domain block by repeating the above steps.

For all the range blocks, we repeat the above procedure to search their matched domain blocks.

4. Simulation results

In order to evaluate the efficiency of the above algorithm, we perform computer simulations on a PC with Pentium 4 3.00GHz CPU. Lena, Peppers and Girl $(512\times 512\times 8)$ are used for evaluating this algorithm. In this experiment, the proposed algorithm combines with Saupe's algorithm [9]. The domain step is 8 pixels and the dimension of feature vector is 16. Consequently, in FGSE, we partition the block to five levels as shown in Fig. 2. For the purpose of comparison, the Saupe's algorithm [9] and the Fisher's algorithm [3] are also implemented.

In Saupe's algorithm, the number M, approximate nearest neighbors returned per search, has an effect on the simulation performance. Thus we study the influence of M on the proposed algorithm and Saupe's algorithm and the result is described in Fig. 3. From this figure, it is obvious that the encoding time of our proposed algorithm is always lower than that of Saupe's and the number is much bigger, our algorithm is much faster.

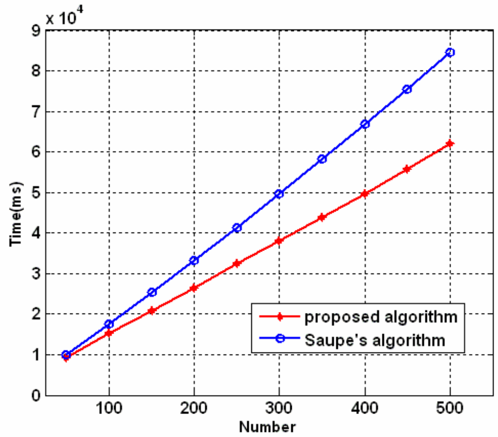

Fig. 3 for the number M of the nearest neighbors returned per search/time comparison on Lena image

And when the number M is 100, the compared results for the three algorithms are listed in Table 1. From this table, it is obvious that the peak signal-to-noise ratio (PSNR) of the proposed algorithm is absolutely the same as that of Saupe's and greater than that of Fisher's algorithm. The speed of the proposed algorithm is more than twenty percent than that of Saupe's and nearly one and a half time as that of Fisher's.

Table 1: Performance of the Proposed algorithm, Saupe's algorithm($M = 100$)and Fisher's algorithm

Image	Proposed algorithm			Saupe's algorithm					Fisher's algorithm				
	PSNR (dB)	Time (ms)	CR	PSNR (dB)	Drop (dB)	Time (ms)	Speed up	CR	PSNR (dB)	Drop (dB)	Time (ms)	Speed up	CR
Lena	28.11	14734	18.36	28.11	0	17203	1.15	18.36	28.03	-0.08	24437	1.40	18.36
Girl	28.56	11532	25.70	28.56	0	14125	1.84	25.70	28.56	0	19985	1.42	25.70
Peppers	28.27	12938	21.42	28.27	0	15656	1.17	21.42	28.21	-0.06	22687	1.43	21.42

Furthermore, we study the number of domain blocks needed to be computed between the proposed algorithm and Saupe's algorithm as $M = 200$ and the result is shown in Table 2. From this table, it is obvious that the matching computation of the proposed algorithm is only half as that of Saupe's. So it greatly decreases the encoding time.

Table 2: Comparison of the number needed compute with Saupe's algorithm and the proposed algorithm ($M = 200$)

comparison results			Lena	Girl	Peppers
Saupe's algorithm	NBNC		7884800	5612800	6777600
	CR		100%	100%	100%
	Time(ms)		32796	27110	30000
	PSNR(dB)		28.11	28.56	28.28
Proposed algorithm	level 0	NRB	0	0	0
		RR	0	0	0
	level 1	NRB	1835778	714148	1586769
		RR	23.3%	12.7%	23.4%
	level 2	NRB	617576	346814	503106
		RR	7.8%	6.2%	7.4%
	level 3	NRB	685391	443549	558440
		RR	8.7%	7.9%	8.3%
	level 4	NRB	753503	551694	618819
		RR	9.6%	9.8%	9.1%
	level 5	NRB	803749	626037	650906
		RR	10.2%	11.2%	9.6%
	NBNC		3188803	2930558	2859560
	CR		40.4%	52.2%	42.2%
	Time(ms)		26016	20313	22532
	PSNR(dB)		28.11	28.56	28.28

NRB: the number of rejected blocks; RR: rejecting ratio;
NBNC: the number of blocks needed compute; CR:computing ratio.

V. Conclusions

This paper presents a fast search algorithm for fractal image coding. First, by using the Cauchy inequality, a relationship between $E(R,D)$ and SAD is derived. Then, using this inequality property, the algorithm can remove many unmatched blocks by FGSE and hence dramatically speeds up the search process in fractal image coding. Simulation results show that the proposed algorithm is betterment for Saupe's algorithm and can combine with other fractal algorithm to improve the encoding speed further.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (No. 90604032, No. 60373028), Specialized Research Fund for the Doctoral Program of Higher Education，Program for New Century Excellent Talents in University and Specialized Research Foundation of BJTU.

References

[1] Barnsley M.F., Sloan A.D., "A better way to compress image". BYTE, Jan.1988, pp.215-223.

[2] A.E.Jacquin, Image coding based on a fractal theory of iterated contractive image transformations, IEEE Trans. Image Processing, Jan.1992, vol.1, pp.18-30.

[3] Y.Fisher, Fractal image compression: Theory and Applications, New York: Springer-Verlag, 1995.

[4] E.W.Jacob, Y.Fisher, R.D.Boss, Image compression: a study of the iterated transform method, Signal Processing 29(1992) 251-263.

[5] C.K.Lee , W.K.Lee, Fast fractal image block coding based on local variance, IEEE Trans. Image Processing, June 1998, vol.7, pp.888-891.

[6] He,C.,Yang,S.X., Huang,X., Variance-based accelerating scheme for fractal image encoding, Electron. Lett., 2004, 40, (2), pp.115-116.

[7] Cheung-Ming Lai, Kin-Man Lam, An efficient algorithm for fractal image coding using kick-out and zero contrast conditions, ISCAS '03 Circuits and Systems, May 2003,Vol 2, pp.25-28.

[8] C.He, S.X.Yang and X.Xu, Fast fractal image compression based on one-norm of normalized block. Electronics Letters, Aug 2004, 40(17), pp.1052-1053.

[9] D.Saupe, Accelerating fractal image compression by multi-dimensional nearest neighbor search, Data Compression Conference, March 1995, pp.222-231.

[10] C.S.Tong, M.Pi, Fast fractal image encoding based on adaptive search, IEEE Trans. Image Processing. Sept. 2001 10(9), pp.1269-1277.

[11] R.Distasi, M.Nappi, A range/domain approximation error-based approach for fractal image compression, IEEE Trans. Image processing Jan.2006 15(1), pp.89-97.

[12] Yao Zhao, Hongxing Wang, Baozong Yuan, Advances in fractal image coding, ACTA Electronica sinica, April 2000, 28(4), pp.95-106.

[13] Ce Zhu, Wei-Song Qi, a new successive elimination algorithm for fast block matching in motion estimation, ISCAS '04 Circuits and Systems, May 2004, Vol.3, pp.23-26.