AN EFFIEIENT ALGORITHM FOR FRACTAL IMAGE CODING USING KICK-OUT AND ZERO CONTRAST CONDITIONS*

*Cheung-Ming Lai, Kin-Man Lam** and Wan-Chi Siu*

Centre for Multimedia Signal Processing
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong

ABSTRACT

In this paper, we propose a fast algorithm for fractal image coding based on a single kick-out condition and the zero contrast prediction. The single kick-out condition can eliminate lots of unmatched domain blocks in the early encoding phase. An efficient method based on the zero contrast prediction is also proposed, which can determine whether the contrast factor for a domain block is zero or not and compute the corresponding difference between the range block and the transformed domain block efficiently and exactly. The proposed algorithm can achieve the same reconstructed image quality as the exhaustive search, and can greatly reduce the required computational complexity. In addition, this algorithm does not need any pre-processing step and additional memory for its implementation, and can combine with other fast fractal algorithms to further improve the speed. Experimental results show that the runtime is reduced by about 50% when compared to the exhaustive search method. The runtime can be reduced by about 75% when our algorithm is combined with the DCT Inner Product algorithm.

1. INTRODUCTION

A significant amount of research work has been done on fractal image compression recently. Fractal image coding can provide a highly reconstructed image quality with a high compression ratio (CR), is independent of resolution, and has a fast decoding process. Fractal theory was first presented by Barnsley, and is based on a mathematical theory called Iterated Function Systems (IFS). Jacquin [1] proposed the first practical fractal image compression scheme that relies on the assumption that image redundancy could be efficiently exploited through self-transformability on a block-wise basis. Fractal image compression is based on the representation of an image by a set of iterated contractive transformations for which the reconstructed image is an approximate fixed point and close to the original image.

The exhaustive search algorithm can obtain the optimal domain block to represent the range block by searching exhaustively all the blocks within the domain pool, but this process suffers from long encoding time and limits its practical applications. To solve this problem, extensive research on fast fractal image encoding algorithms [2-4] has been carried out. However, these techniques reduce the required computation at the expense of additional memory and degradation of the reconstructed image quality.

In this paper, we propose an efficient algorithm based on a single kick-out condition and the zero contrast prediction, which can greatly reduce the required computation as compared to the exhaustive search, while maintaining the same reconstructed image quality. Our proposed kick-out condition can determine efficiently whether a domain block is a good representation of a range block, and so excessive computation can be avoided in the early stage. With zero contrast prediction, the computation involved is further reduced. Moreover, the algorithm dues to not require any pre-processing and extra memory for its implementation. Our proposed approach can also be combined with other fast encoding methods to further speedup the encoding time. Experimental results show that the runtime can be reduced by about 75% when our algorithm is combined with the DCT Inner Product algorithm [5].

This paper is organized as follows. We briefly describe the fundamentals of the fractal coding algorithm and review the DCT Inner Product approaches in Section 2. Section 3 presents our new fractal image compression algorithm based on a kick-out condition and the zero contrast prediction. In Section 4, we compare the performance of our proposed fast algorithm, the full search, the DCT Inner Product algorithm, and our algorithm combined with the DCT algorithm. Finally, conclusions are given in Section 5.

* This research work was supported by the HKPolyU research grant G-W088.

** E-mail: enkmlam@polyu.edu.hk.

0-7803-7761-3/03/$17.00 ©2003 IEEE

2. REVIEW OF FRACTAL IMAGE COMPRESSION

In the fractal image compression scheme, an image, f_{orig}, of size $r\times r$ is partitioned into two basic block units: the range block and the domain block. The range blocks are a set of non-overlapping image blocks of size $k=n\times n$, which are denoted as $\{R_i\}_{i=1}^{N_R} = \{r_{i1}, r_{i2}, \cdots, r_{ik}\}_{i=1}^{N_R}$. The number of range blocks is $N_R = (\frac{r}{n})\times(\frac{r}{n})$, and image f_{orig} is a union of the range blocks, $\{R_i\}_{i=1}^{N_R}$:

$$f_{orig} = \bigcup_{i=1}^{N_R} R_i \qquad (1)$$

Overlapping image blocks of f_{orig} in a domain pool with size larger than that of the range blocks are called domain blocks. A collection of all these domain blocks form a domain pool. These domain blocks can be obtained by sliding a window of size $l = m\times m$, where $m > n$, throughout the image to construct the domain pool. The size of a domain block is usually four times that of a range block, i.e. $l = 2n\times 2n$. To encode a range block R, each of the blocks in the domain pool is scaled to the size of the range block, and is then compared to R with respect to intensity offset and contrast parameters, as well as the isometry transformations. The set of contracted domain blocks is denoted as $\{D_i\}_{i=1}^{N_D} = \{d_{i1}, d_{i2} ..., d_{il}\}_{i=1}^{N_D}$, where N_D is the number of domain blocks in the domain pool. The corresponding parameters for the affine transformation τ are determined by minimizing the following equation:

$$E(R, D_i) = \|R - (s.D_i + oI)\|^2 \text{, where } o, s \in R \qquad (2)$$

In this equation, D_i is the contracted domain block under an isometry transformation, I denotes a unity vector of dimension k, s and o are the contrast and offset parameters, respectively. For a given range block and the corresponding domain block, these two parameters are given as follows:

$$s = \frac{\langle R,D\rangle - \frac{1}{k}\langle R,1\rangle\langle D,1\rangle}{\|D\|^2 - \frac{1}{k}\langle D,1\rangle^2} \text{ and}$$

$$o = \frac{1}{k}\left(\langle R,1\rangle - s\langle D,1\rangle\right) \qquad (3)$$

The $\| \ \|$ is the two-norm and $\langle .,. \rangle$ is inner product. The contrast factor should be $-1 < s < 1$ to ensure the contractivity of the transformation. The domain block which results in the smallest difference with equation (2) is then chosen as the best matched block, and the corresponding parameters for the transformations $\{\tau_i \mid i = 1,2,\ldots,N_R\}$ are encoded and stored. At the decoding phase, the transformation parameters are recursively applied to an arbitrary initial image, which will then converge to the fractal image after fewer than 10 iterations.

Truong *et al* [5] proposed an efficient image coding algorithm which can produce the same image quality as exhaustive search. In this method, the image blocks are first demeaned, and the error function (2) between a range block and a transformed domain block can be simplified as follows:

$$E(R,D) = \|R - \bar{r}I\|^2 - \frac{\langle R - \bar{r}I, D - \bar{d}I\rangle^2}{\|D - \bar{d}I\|^2} \qquad (4)$$

where $\bar{r}$ and $\bar{d}$ are the means of the range block and domain block, respectively, while I is a vector with all 1's and of the same dimension as R and D. The most computational part of (4) is the inner product $\langle R - \bar{r}I, D - \bar{d}I\rangle$. To determine the best matched domain block for a range block, the isometry transformation consists four orientations and four reflections of each domain block. In other words, the error function has to be computed eight times; once for each of the transformed domain blocks. Most of the computational cost of this method comes from the overhead for calculating the inner product of the range block and the transformed domain block in computing the error function. In order to reduce computation involved, [5] proposed using Discrete Cosine Transform (DCT) to convert the image block to the frequency domain, which can reduce the number of computations of the inner product from eight to two. The other inner products can be obtained by a proper arrangement of these two inner products.

3. PROPOSED ALGORITHM

Our algorithm uses a kick-out condition in searching for the best matched domain block to represent a range block. Those domain blocks in the domain pool satisfying kick-out condition will be bypassed, so no further computations will be needed. For this kick-out condition, we first convert the full search equation (2) from two parameters, i.e. the contrast s and the offset o, to a function which only contains the contrast s. Based on this formulation, we can successively eliminate the search space in the domain pool and thus decrease the computation required to compare a range block and a transformed domain block. To further reduce the computation, we propose a simple method to determine the zero contrast condition. When this condition occurs, the range block can be coded without performing any range-domain block matching.

3.1 The Kick-out Condition

From equation (2), the error function can be further simplified as follows:

$$E(R,D)=\left[\|R\|^2-\tfrac{1}{k}\langle R,1\rangle^2\right]-\frac{\left[\langle R,D\rangle-\frac{1}{k}\langle R,1\rangle\langle D,1\rangle\right]^2}{\left[\|D\|^2-\frac{1}{k}\langle D,1\rangle^2\right]}$$

$$=\left[\|R\|^2-\tfrac{1}{k}\langle R,1\rangle^2\right]-\frac{\left[\langle R,D\rangle-\frac{1}{k}\langle R,1\rangle\langle D,1\rangle\right]^2}{\left[\|D\|^2-\frac{1}{k}\langle D,1\rangle^2\right]^2}\left[\|D\|^2-\tfrac{1}{k}\langle D,1\rangle^2\right] \quad (5)$$

As $s=\dfrac{\langle R,D\rangle-\frac{1}{k}\langle R,1\rangle\langle D,1\rangle}{\|D\|^2-\frac{1}{k}\langle D,1\rangle^2}$, the error function $E(R,D)$ can be written as follows:

$$E(R,D)= A-s^2B \quad (6)$$

where $A=\|R\|^2-\frac{1}{k}\langle R,1\rangle^2$ and $B=\|D\|^2-\frac{1}{k}\langle D,1\rangle^2$.

The coefficient s is limited to the range (-1,1) to ensure convergence in the decoding process. If A is greater than B, the maximum error occurs when s=0 while minimum error occurs when s=1. Therefore, we have:

If $A-B \geq 0$, then

1) The maximum error occurs when s=0,

$$E_{\max}= A-s^2B=A \quad (7)$$

2) The minimum error occurs when s=1,

$$E_{\min}=A-s^2B=A-B \quad (8)$$

This means that, in finding the best matched domain block, the search is performed only if the minimum error $E_{\min}$ for the domain block under consideration is less than the current minimum error $d_{\min}$. Thus, we propose the kick-out condition as follows:

$$E_{\min}=A-B\geq d_{\min} \quad (9)$$

Based on (9), we propose a fast search algorithm which can reject dissimilar domain blocks efficiently for a given range block. In our algorithm, we select the domain blocks from left to right and top to bottom. The first domain block D_1 is considered to be the initial best matched domain block. The current minimum distance $d_{\min}$ is set to the distortion $E(R,D_1)$ and the search proceeds in the raster scan order. To determine whether the next candidate domain block D_2 is closer to R than the current best match D_1, we compute $E_{\min}(R,D_2)$ and compare it to $d_{\min}$. If $E_{\min}(R,D_2)$ is larger than or equal to $d_{\min}$, it also means that the condition $E(R,D_2)\geq d_{\min}$ is always guaranteed. The domain block D_2 is therefore rejected. Otherwise, the actual distortion $E(R,D_2)$ is calculated and compared to $d_{\min}$. If $E(R,D_2)\geq d_{\min}$, D_2 is rejected for the same reason mentioned above. Otherwise, $d_{\min}$ is replaced by $E(R,D_2)$ and the current best matched domain block is set to D_2. This process is repeated for all the domain blocks D_i in the domain pool to find the best matched one for an input range block. Based on this kick-out condition, the required computation for searching the best matched domain block will be greatly reduced.

3.2 Fast Error Calculation using Zero Contrast Prediction

In the implementation, the contrast factor s is encoded using 5 bits. Therefore, any value of the contrast s falling within (–0.03125, 0.03125) will be set to zero after quantization. With (6), as $|s|<1$, this means that the zero contrast condition will happen only when $A<B$:

$$E(R,D)=A-s^2B\geq 0$$

$$\text{or } \sqrt{\frac{A}{B}}\geq |s| \quad (10)$$

The contrast factor s is quantized to 0 if the absolute value of s is less than 0.03125, i.e.:

$$0.03125>\sqrt{\frac{A}{B}}\geq |s| \quad (11)$$

When s is set to zero, the corresponding error is given as follows:

$$E(R,D)=A \quad (12)$$

In this case, the range block can be encoded without performing any range-domain matching, and their error can be represented by the constant value A.

3.3 Combining Other Approaches

Our proposed algorithm can combine with other fast fractal algorithms to further improve their speed. One example is the DCT Inner Product [5] approach, which allows the computation of two inner products only for the four orientations and four reflections of a domain block. However, the whole domain pool still has to be considered in order to obtain the best matched domain block. This DCT approach can combine with our algorithm to further improve its speed. In encoding an image, the single kick-out condition (9) will be checked to reject those dissimilar domain blocks. Then, zero contrast prediction (11) is used to determine whether the contrast factor is zero or not, and the corresponding error function can be computed without performing the range-domain block matching. Therefore, the required runtime for the algorithm can be further reduced.

4. EXPERIMENTAL RESULTS

In the experiment, a three-level quadtree partition scheme with range block sizes of 4×4, 8×8 and 16×16 pixels, and a search grid of one are used. Three popular 512×512 images, Lena, Boat and Goldhill, are used to evaluate the performance of the proposed algorithm and other algorithms. The computer used is a Pentium III 500MHz. The runtimes (in second) for our proposed algorithm and full search are listed in Table 1. We measured the runtimes of our algorithm based on (i) the single kick-out condition (i.e. case 1), (ii) the zero contrast prediction (i.e. case 2), and (iii) the combination of both conditions (i.e. case1 and 2). Experimental results show that our proposed algorithm considering both conditions can reduce the required computation by about 50% as compared to the exhaustive search method. In other words, a large number of domain blocks are rejected for performing the range-domain block matching by the kick-out condition, and a number of the error functions are obtained based on the zero contrast prediction. Experimental results show that about 52 % of the domain blocks are rejected by the kick-out condition, while 6% of the remaining range-domain block matching can use the zero contrast prediction to compute the corresponding error functions. This means that the computational complexity can be reduced by more than half using our proposed algorithms. In (6), the kick-out condition has not considered the effect of quantizing the luminance offset, so we cannot guarantee that the optimal domain block will be obtained. Therefore, in order to obtain the best domain block for representing a range block, we set a tolerance of 10% more when comparing the minimum error between a range block and a domain block of the current minimum error d_{min}. With this setting, we found that the reconstructed image quality will be equal to the exhaustive search.

The performance of our algorithm combined with the DCT Inner Product approach was also investigated. The size of the range blocks is set to 8×8 only. We combined the DCT approach with the single kick-out condition and the zero contrast prediction. These combined algorithms were compared with the baseline method and the DCT Inner Product method [5] in terms of the encoding time and PSNR. The experimental results are tabulated in Table 2, which shows that the runtime of the combined algorithm is about 25% of the baseline approach and 50% of the DCT approach. Furthermore, the PSNR based on our algorithm is the same as that of the baseline method.

4. CONCLUSIONS

In this paper, we propose a single kick-out condition and the use of zero contrast prediction to speedup the encoding process. The efficiency of the kick-out condition depends on how quickly the global minimum error is detected. Once this global error is found, most of the remaining domain blocks will be rejected and range-domain block matching will not be performed. Experimental results show that the runtime of our algorithm is about 50% of the exhaustive search. Our algorithm can also be combined with other fast fractal coding algorithms, such as the DCT Inner Product, to further improve the speed. The combined algorithm can reduce the required computation by about 75% as compared to the baseline approach.

5. REFERENCES

[1] A. E. Jacquin, "Fractal image coding: A review," Proc. IEEE, vol. 81,pp. 1451–1465, Oct. 1993.

[2] Y. Fisher, Ed., Fractal Image Compression: Theory and Application. Berlin, Germany: Springer-Verlag, 1995.

[3] C. K. Lee and W. K. Lee, "Fast fractal image block coding based on local variances," IEEE Trans. Image Processing, vol. 7, pp. 888–891, June 1998.

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

[5] T. K Truong, J. H Jeng, I. S. Reed, P. C. Lee, and A. Q. Li, "A Fast Encoding Algorithm for Fractal Image Compression Using the DCT Inner product". IEEE Trans. Image Processing, vol. 9,No. 4, pp. 529–535, April 2000.

Test Images		Algorithms	
		Full Search	Proposed Algorithm (Case 1 & 2)
Lena	Time (s)	5940	3242
	PSNR (dB)	34.91	34.91
	CR	15.50	15.50
Boat	Time(s)	8463	3666
	PSNR (dB)	34.91	34.91
	CR	9.71	9.71
Goldhill	Time(s)	8976	5755
	PSNR (dB)	33.36	33.36
	CR	9.11	9.11
Peppers	Time(s)	5835	3101
	PSNR (dB)	34.15	34.15
	CR	16.48	16.48

Table 1. Comparison of the coding results using domain grid of one and three level quadtree partitioning (16×16, 8×8 and 4×4).

Test Images		Algorithms			
		Baseline	DCT	DCT + Proposed Algorith m case (1)	DCT + Proposed Algorithm case (1) & (2)
Lena	Time (s)	5340	2473	1540	1342
	PSNR (dB)	31.14	31.14	31.14	31.14
Boat	Time (s)	5340	2473	1571	1376
	PSNR (dB)	29.21	29.21	29.21	29.21
Goldhill	Time (s)	5340	2473	1625	1501
	PSNR (dB)	29.68	29.68	29.68	29.68
Peppers	Time (s)	5340	2473	1521	1319
	PSNR (dB)	34.11	34.11	34.11	34.11

Table 2. Comparison of the coding results using domain grid of one and 8×8 range block.