Fast Fine Granularity Decoding of Fractal Image Coding

CHEN Yisong, ZHANG Fuyan
(State Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing, 210093)
Email: yschen@graphics.nju.edu.cn, fyzhang@nju.edu.cn

ABSTRACT

Iterative decoding method for Fractal image coding has good "scalability" performance. Based on the nature of fractal image coding, a pixel update algorithm along with single-buffer mechanism is proposed in place of mapping update algorithm using double-buffer mechanism, which effectively saves memory expenses and realize scalable decoding in finer granularity. Also an improved block ordered decoding method is put forward and used in variable block size based fractal image coding, leading to a faster convergence speed.

Index terms: fractal image coding, iterated function system.

1 INTRODUCTION

One of the most important characteristics of fractal image coding is its unsymmetric property of encoding and decoding processing[1][2][3]. Coding time is rather long for domain codebook generation and domain/range matching operation, while decoding algorithm is relatively simple and fast. This makes it a good candidate for those storage and retrieval applications where coding processing is performed only once but decoding processing needs doing many times.

"Resolution independence" has been claimed as one of the main advantages of fractal image compression[4]. A data stream produced by an encoder of fractal image coding can be reconstructed in different spatial resolution with a fast pyramidal decoding algorithm[5].

Rebuilt of the encoded image is achieved by computing the fixed point of the image transform T using iterated function system. Every iterating of the image is an approach to the final attractor. In this view, The iterating process actually also gives a quality scalable decoding. Speeding up convergence of the decoding algorithm is one of the important branches of fractal coding. Some algorithms are fast mapping updating, ordered iteration, combination with VQ, etc. [6][7][8]

Double-buffer mechanism is commonly used in nowadays decoding algorithm. However, it is shown in this paper that a single-buffer algorithm is much more advisable in terms of computation and memory saving. With single-buffer algorithm, a fine granularity iterative decoding in which one single mapping is seemed as a minimal iteration unit, is easily accomplished. Additionally, a variable size block ordered decoding algorithm is addressed and used in decoding schemes based on variable size block segmentation, Resulting in a faster convergence speed.

The outline of this paper is as follows: section 2 offers a brief review of some work aiming at fast decoding of fractal coded image and gives a simple description of mapping update decoding and ordered block decoding. Section 3 proposes a single-buffer decoding algorithm and construct a fine granularity iterative decoding model based on single mapping. Also a variable size block ordered decoding is detailed in this section. Section 4 presents simulation results of the new decoding model. The paper concludes with section 5, where some perspective is given on the new results introduced in this work.

2 FAST DECODING

In conventional fractal coding, the image $x^* \in R^n$ can be coded as an approximation of a unique fixed point x_T of an affine mapping T defined by

$$T(x) = Ax + b \qquad (1)$$

where A is a real n*n matrix and b is a real column vector of order n. The reconstruction of the image by the decoder proceeds by iteratively applying T to any arbitrary initial image x_0. If the spectral radius $r(A)$ of the matrix A is less then one, then the sequence of iterates $\{T^k(x_0)\}_k$ converges to the attractor of the iterated system defined by (2)

$$x_T = (I_n - A)^{-1} b \qquad (2)$$

where I_n is the identity matrix of order n. Here the vector $T^k(x)$, which is also denoted by $x^{(k)}$ is defined as (3)

$$x^{(k+1)} = T(x^{(k)}) = Ax^{(k)} + b \quad (3)$$

In actual coding applications, the mapping T is the union of several sub-mappings. An optimal domain/range mapping is found for each nonoverlapping range block and the corresponding coefficients are recorded as codeword.

When decoding, the mapping coefficients are parsed and the iteration is done from an arbitrary initial image. One mapping in such a computing construction is as equation (4).

$$x_{r,u}^{(k+1)} = A_u x_{d,u}^{(k)} + b_u \quad (4)$$

$x_{r,u}^{(k+1)}$ is the range vector of the u-th mapping in the (k+1)-th image iteration, $x_{d,u}^{(k)}$ is the domain vector of the u-th mapping after the k-th image iteration, and A_u is the relative mapping coefficients matrix.

Two image buffers are needed in conventional decoding algorithms. Mapping operation sequentially maps the updating image buffer to the updated image buffer. After an integral iteration of the whole image, as the preparation of next iteration, the updating buffer and the updated buffer is swapped.

The decoding process can be accelerated using mapping update algorithm [7], The basic idea of the algorithm is that the range block can be updated immediately after the mapping process. In mapping update altorithm formula (4) can be rewritten as formula (5)

$$x_{r,u}^{(k+1)} = A_u^{(k+1)} x_{d,u-}^{(k+1)} + A_u^{(k)} x_{d,u+}^{(k)} + b_u \quad (5)$$

$x_{d,u-}^{(k+1)}$ means pixels in the u-th domain vector of (k+1)-th mapping that have been updated in former mappings of the same iteration, while $x_{d,u+}^{(k+1)}$ means pixels in the u-th domain vector of (k+1)-th mapping that have not been updated in former mappings. $A_u^{(k)}$ means dynamically updated coefficients matrix.

If the mapping order is properly selected and some "important mappings" are given higher priority, The convergence can be further sped up. An ordered decoding algorithm is put forward in reference [8] based on this idea.

However, still some defects exist in above mapping update algorithm and ordered algorithm. Double-buffer mechanism, when used in mapping update algorithm as in [6], means that one more buffer updating operation is needed after each mapping, which is a contraction to the original goal of speeding up the convergence. The ordered decoding algorithm in [8] is realized only in a fixed uniform segmentation and fixed size block fractal coding scheme, not suitable for more flexible variable size block based coding schemes. A perspective solution to above two problems is thoroughly discussed in section 3.

3 FINE GRANULARITY DECODING

The instant updating method described by formula (5) can go further. A complete parallel notion is as formula (6).

$$p_{r,u,m}^{(k+1)} = S_u^{(k+1)} p_{d,u,m-}^{(k+1)} + S_u^{(k)} p_{d,u,m+}^{(k)} + b_u \quad (6)$$

$p_{r,u,m}^{(k+1)}$ means current computing pixels in the u-th range vector of (k+1)-th mapping. $p_{d,u,m-}^{(k+1)}$ means the pixels in the u-th domain vector of the (k+1)-th mapping that have been updated in former and CURRENT mappings of the same iteration. $p_{d,u,m+}^{(k)}$ means the pixels in the u-th domain vector of (k+1)-th mapping that have not been updated in former and CURRENT mappings. $S_u^{(k)}$ means relative dynamically updated coefficients row vector.

A comparison of equation (5) and (6) shows that (5) describes an image updating algorithm in block mapping level, while (6) describes an image updating algorithm in deeper pixel level. We call it pixel update algorithm in this paper.

Obviously, in pixel update algorithm pixels get updating in a finer layer. Thus a still faster convergence is expected. This can be further explained using figure 1. In which a spatial probability distribution of relative position of optimally matched domain and range blocks is shown[9]. The ordination (b_x, b_y) means relative distance of center position of matched domain and range blocks. It can be seen from the figure that matched domain and range blocks always near each other. This fact is the main reason why pixel update algorithm results in faster convergence speed.

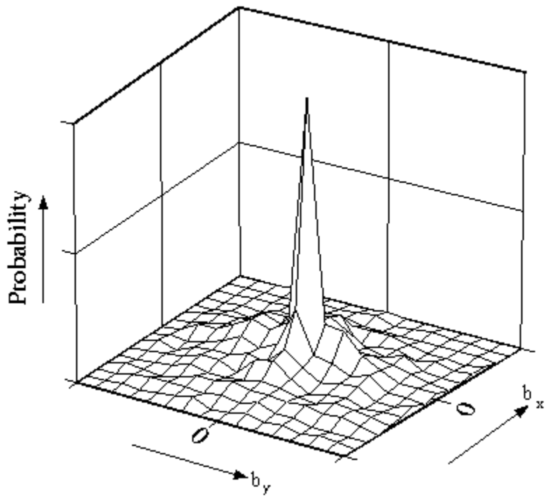

Figure 1 probability density for offset vector of domain/range blocks

Pixel update algorithm makes possible single-buffer mechanism in decoding without extra computing expense. One image buffer is sufficient for the decoder, which act as both updating buffer and updated buffer mentioned above. The mapping is directly done in the buffer. thus, also pixel update algorithm saves one buffer updating operation in every loop execution, which is the main disadvantage of mapping update algorithm.

The notion can be extended more. In conventional fractal decoding, the whole image is used as a minimal unit of iteration, which we call coarse granularity decoding. In fact, this "minimal unit" can be further fined to a single mapping, which we call fine granularity decoding. In coarse granularity decoding, the number of iteration of the whole image is used as the parameter controlling the finish of decoding. The parameter is replaced in fine granularity decoding by the number of single mapping, thus in some sense realizes "continuous scalability".

The decoding algorithm with such a scalability feature can be further optimized aiming at getting optimal reconstructed image after arbitrary given mapping times. This can be implemented by sorting all the mappings and gives those more "active" pixels more opportunities to take part in mapping computation. No doubt this is to the good of fast decoding processing.

The ordered decoding algorithm in [8] is based on uniform segmentation, fixed size block fractal image coding and there are position restrictions between domain and range blocks. So this algorithm is not suitable for more flexible coding methods based on adaptive image segmentation and variable size block. Nevertheless, An effective ordering is still possible in such methods. Actually, the fact that block size is variable itself hints a rather attracting sorting scheme. Clearly, to achieve an efficient convergence, former mappings should finish as much pixel updating as possible. This opinion leads to the idea of descendant sorting all the mappings by their range block size. The sorting can be performed either in encoder or in decoder at the same time of generating or parsing mapping coefficients using simple data structure without increasing of computational cost.

4 EXPERIMENTAL RESULTS

Four different decoding schemes are tested in our experiment. They are conventional iterative decoding(scheme 1), mapping update decoding(scheme 2), pixel update decoding (scheme 3) and block ordered pixel update decoding(scheme 4). Scheme 3 and scheme 4 are respectively based on above metioned fine granularity decoding idea and variable size block sorting idea. The input stream is generated by a common encoder, which uses adaptive quadtree-based image segmentation algorithm[10]. The results are given in table 1. We can see from the table that the main distinction of the four algorithms is the convergence speed of the algorithm in the first three iterations. The decoding schemes using update algorithms(scheme 2,3,4) have faster execution speed than conventional scheme (scheme 1), with a PSNR gain of at least 2~3 dB. Of the three schemes using update algorithm, block ordered algorithm (scheme 4) outperforms the other two(scheme 2,3) with a PSNR gain of 0.8~2 dB. For the reconstructed images after the same times of iterations, scheme 3 is a bit better than scheme 2, this is because scheme 3 makes more subtle local processes in the iteration loop. Another merit of scheme 3 superior to scheme 2 is that the single-buffer mechanism saves one buffer's usage and omits frequent buffer updating, which leads to a time saving of about 5%, as shown in the table. These distinction of scheme 3 and scheme 2 comes from the intrinsic nature of mathematical principle of iterated function system in the two schemes.

Table 1 results of four different decoding schemes

	Scheme 1		Scheme 2		Scheme 3		Scheme 4	
Iteration	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR
1	0.19s	19.97	0.21s	21.56	0.20s	21.63	0.20s	22.99
2	0.34s	24.36	0.36s	27.61	0.35s	27.78	0.35s	29.72
3	0.48s	28.45	0.52s	31.61	0.50s	31.77	0.50s	32.51
4	0.63s	31.51	0.68s	32.72	0.65s	32.76	0.65s	32.86
5	0.76s	32.63	0.84s	32.89	0.80s	32.90	0.80s	32.91

Scheme 4 is an improvement of scheme 3 based on the algorithm of variable size block ordering, which partly realizes the idea of "continuous quality scalable decoding". That is, single mapping is used as minimal unit in decoding algorithm. In such an execution unit, a proper mapping is selected by a given principle to get an optimal rate-distortion feature. Decoding can finish after any arbitrary times of mapping, leading to an at least suboptimal reconstructed image account for the number of executed mappings.

This notion is very similar to the "scalability" notion in EZW image coding, although there is an essential distinction between them. In EZW coding, "scalability" refers to that a code stream can be truncated at any point and get a suboptimal reconstructed representation in such a code rate, while the "scalability" in this paper is accomplished through truncating not code stream but decoding process.

5 CONCLUSION

"Continuous quality scalable decoding" notion of fractal image coding proposed in the paper has the same kernel idea as scalability in a common sense. This paper has given a simple implementing model of such a scalable property. Although the notion of fine granularity is attracting and has been partly realized in the model, It does not really break out the framework of conventional coarse granularity decoding, for the fact that mappings are still executed sequentially in a loop structure. If we can find a criterion which efficiently judges the inner importance of each mapping in the mapping set, then select an proper probability analysis model to adaptively generate current optimal mapping in decoding process, A fine granularity decoding in a real sense can be expected.

6 REFERENCES

[1] M. F. Barnsley, Fractals Everywhere, Academic Press, San Diego, 1988

[2] A. E. Jacquin, Image coding based on a fractal theory of iterated contractive image transformation, IEEE Trans. on Image Processing, 1(1), January 1992, pp18-30

[3] B. Wohlberg and Gerhard de Jager, A Review of Fractal Image Coding Literature, IEEE Trans. Image Processing, Vol 8, No.12, December, 1999, pp1716-1729

[4] Y. Fisher, Fractal Image Compression--Theory and Application, Springer-Verlag, New York, 1994.

[5] Z. Baharav, D. Malah and E. Karnini, Hierarchical Interpretation of Fractal Image Coding and its Application to Fast Decoding, Proceedings of Intl. Conference on Digital Signal Processing, Cyprus, July 1993, pp190-195

[6] R. Hamzaoui, A new decoding algorithm for fractal image compression, Electronics Letters, 32,14 (1996) 1273-1274.

[7] R. Hamzaoui, Fast decoding algorithms for fractal image compression, Technical Report 86, Institut für Informatik, University of Freiburg, 1997.

[8] R. Hamzaoui, Ordered decoding algorithm for fractal image compression, Proc. PCS'97 International Picture Coding Symposium, pp91-95, Berlin, Sept. 1997.

[9] B. Hürtgen and C. Stiller, Fast hierarchical codebook search for fractal coding of still images, Proceedings EOS/SPIE Visual Communications and PACS for Medical Applications '93 Vol. 1977 Berlin, Germany pp. 397-408, 1993.

[10] M. Polvere, M. Nappi, Speed-up Methods in Fractal Image Coding: Comparison of Methods, to appear in IEEE Transaction on Image Processing.