Feature Difference Classification Method in Fractal Image Coding

Chen Yisong, Wang Guoping, Dong Shihai
HCI & Multimedia Lab of Peking University, Peking, 100871
Email:cys@graphics.pku.edu.cn

Abstract: In this paper, we present a classification algorithm in fractal image coding. Based on the contraction characteristics of transformations in fractal image coding, the algorithm uses the notion of feature difference to speed up the domain-range matching routine of the coding. The algorithm can effectively exclude pseudo matches during the process of domain-range matching and result in significant improvement of the rate-distortion performance. It can also be easily realized in cooperation with many other speedup schemes.
Keywords: Fractal image coding, Classification, Domain-range match.

1.Introduction

Fractal image coding[1][2][3][4] has many good properties such as high compression ratio, rapid decoding and resolution independence. However, the complex domain block calculation and domain-range matching process are usually time-consuming. Block classification[3] and split decision functions[5] are often used to speed up the coding process. Block classification is by far the most widely used. In block classification algorithm, the original image is partitioned into nonoverlapped range blocks and overlapped domain blocks. All the domain blocks are classified into several classes based on some predefined block features. Only domain and range blocks with the same feature take part in domain-range matching job. Block classification can lead to substantial computational cost saving without much fidelity loss. Some commonly used classification schemes are Fisher[3], Hurtgen[6], Saupe[7], Polvere[8], etc.

D. Saupe's classification scheme seems to give the best rate-distortion performance among all above schemes in the same computational complexity level. However, the algorithm is always bothered with the problem that more time is needed for higher rebuilt image quality, which conflicts with the original goal of classification. In this paper, we address a method using the notion of feature difference in Saupe classification. This method can dramatically simplify the matching process in fractal coding without much fidelity loss, or, conversely, improve rebuilt image quality with the same or even shorter coding time.

The rest part of the paper is outlined as follows. Section 2 gives a brief review of the famous Saupe classification method. In section 3, we point out one main shortage of Saupe method and introduce a feature difference decision rule to overcome the drawback. Experimental results given in section 4 show that the new algorithm is very effective. Finally in section 5, we conclude and summarize the paper.

2. A review of Saupe classification method

In fractal image coding algorithms based on affine transformation, for a given range block Z, the target is to find a proper domain vector X_i in the predefined domain pool to satisfy the following equation:

$$E(x_i, z) = \min_{a,b\in R} \left\| z - (ae + bx_i) \right\| \qquad (1)$$

where

$$e = \frac{1}{\sqrt{d}}(1,1,\ldots 1) \in R^d \qquad (2)$$

D. Saupe provided an attractive classification algorithm to solve the above problem. Detailed principles and applications of Saupe classification have been reported in previous literature[7][8]. Here we just indicate that the key idea of Saupe method is that the better-matched domain and range blocks, when transformed to a feature vector space, are expected to have smaller distance. Hence, the domain-range match problem is converted to nearest neighbor searching job in the feature space, which can be easily solved with some mature algorithms such as a kd-tree approach [9][10].

In rectangle block based fractal image coding, Saupe method is often used along with the isometric computation technique of Fisher[3]. In this scheme, all domain and range blocks are transformed to one of the three major class forms using one of the eight isometric transformations. This preprocessing step speeds up the match time 8 times in theory. Again, in this way, it is only needed to consider transformations with scale parameter b>0. It is reasonable in that a 180-degree rotation changes the sign of the parameter b without destroying of linear correlation.

3. Feature difference decision criterion

In most current implements of Saupe classification, for a given range block, nearest neighbor searching is used to find m suboptimal solutions in the feature space. m is called match parameter in this paper. Then linear regression analysis is used to find the best one from the m candidates. If m takes a large value, the search is sufficient and quite nice fidelity can be obtained. However, it usually costs a very long searching time. On the other hand, if m takes a small value, although matching job is accelerated, it correspondingly increases the risk of finding unsuitable matches, which leads to a rapid decrease of rebuilt image quality. This problem is described in Table 1. As is seen in the table, matching time decreases accordingly with the decrease of m, but PSNR value also decreases rapidly.

Table 1. Affection of the value of m to matching time and PSNR

M	2	5	10	20	50
Time	4.78	5.33	6.26	7.85	11.92
PSNR	32.07	33.20	33.52	33.66	33.70

Can a higher rebuilt image quality be obtained with a smaller match parameter value? First, we give the mathematical expression of the feature vector V of a rectangle block B with n pixels

$$V = (B - E(B)) / \sqrt{nD(B)} \quad (3)$$

Formula (3) shows that the feature vector of the block is in fact just a linear transformation of the original block vector, which gives smaller scale parameters to blocks with bigger variance, so that blocks with better linear correlation will have smaller distance in the feature space. Note that the parameter b in fomula (1) calculated with this method can be any nonzero constant, while an actual affine transformation in fractal coding requires that |b|<1. In fact, this property was completely lost in a conventional Saupe classification algorithm. For instance, if m is very small, the optimal nearest neighbour often leads to one b with value greater than 1. Conveniitional algorithms usually compute the value of b and do a truncation if |b|>1. This method not only loses computation precision but also wastes computation time.

From above analysis, it is obvious that conventional Saupe scheme neglects one basic property of fractal image coding, i.e. the contraction property of the affine transformation. If the property is properly employed in the coding process, much better rebuilt image quality can be obtained using relatively smaller m value with the similar coding speed; or, reversely, a equivalent PSNR value can be achieved with much shorter coding time. Therefore, We propose the following notion of "feature difference" based on above ideas.

For a single pixel B_i in some domain block D, affine transformation in fractal coding maps its graylevel value to the range block R using the following formula:

$$R(B_i) = sD(B_i) + o \quad (4)$$

where the scale factor s satisfies |s|<1. If Fisher method is used as a preprocessing method, we can further assume that 0<s<1.

Suppose that the maximum and minimum of the pixels' graylevel values in the block are respectively B_{max} and B_{min}. We define the feature difference of the block using the following formula:

$$Diff(B) = B_{\max} - B_{\min} \quad (5)$$

So, for a match between the domain block Bd and the range block Br, use (4) (5) and we have

$$\begin{aligned} Diff(Br) &= B_{\max} - B_{\min} \\ &= s(Bd_{\max}) + o - (s(Bd_{\min}) + o) \\ &= s(Bd_{\max} - Bd_{\min}) \\ &= sDiff(Bd) \end{aligned} \quad (6)$$

Considering contraction requirement $0 < s < 1$, we now get

$$Diff(Br) \le Diff(Bd) \quad (7)$$

formula (7) provides a convenient and effective decision rule of precluding improper domain-range match, i.e. if one candidate domain block found by Saupe method has a smaller feature difference than the target range block, the domain block can be safely precluded out of the linear regression computation, and unnecessary computation time will be greatly reduced.

Situation in actual implements is a little different. Research work shows that a bigger threshold of the scale factor s in (4) always leads to a better coding effect [11]. This theory also works for our decision criterion (7). A new improved decision criterion looks like the following formula (8):

$$Diff(Br) \leq s \cdot Diff(Bd) \quad (8)$$

where s is an empirical coefficient that can be a little greater than 1.

With such a decision rule, for a predefined threshold, only domain blocks with feature difference satisfying (8) need further regression analysis while others are all discarded. From (4) it is obvious that the calculation of the feature difference is fairly simple and adds no complexity to the whole algorithm. Since many "pseudo matched blocks" are effectively precluded, a higher rebuilt image quality or a shorter coding time can be expected for different application requirements.

4. Experimental results

The most broadly used quadtree partition is employed in our experiments, which test the following three classification algorithms:

a. Fisher-Saupe classification with larger match parameter (PSNR-oriented or P-S method).

b. Fisher-Saupe classification with smaller match parameter (Time-oriented or T-S method).

c. Fisher-Saupe classification using feature difference decision criterion (D-S method).

The only difference of P-S method and T-S method is the value of the match parameter. From the analysis at the beginning of section 3, it is clear that P-S method can result in higher rebuilt image quality and T-S method can result in shorter coding time under the same experimental environment. The three methods are each used to encode the 512x512 graylevel Lena image. In D-S method, coefficient s in formula (8) takes 1.25, which is a good empirical value for most natural images. All coding processes are mainly composed of 2 steps. In the first step, all domain blocks are extracted from the original image and some preprocessing computations are done. Time used in this step is called preprocessing time. In the second step, the original image is partitioned recursively and domain-range match job is done to obtain coding parameter. Time used in this step is called matching time. The sum of preprocessing time and matching time is called coding time.

Table 2. D-S method compared with P-S and T-S method

	P-S(m=50)	T-S(m=5)	D-S(m=5)
Coding time	22.74	14.78	14.29
Matching time	12.41	5.33	4.89
CR	17.22	16.14	16.71
PSNR	33.70	33.20	33.71

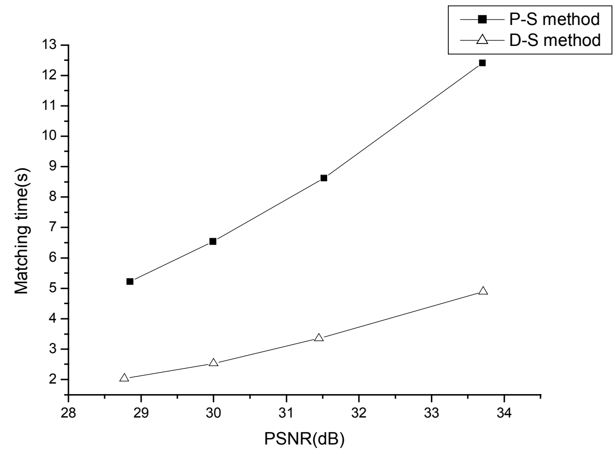

Figure 1. PSNR-Matching time curves of P-S and D-S method

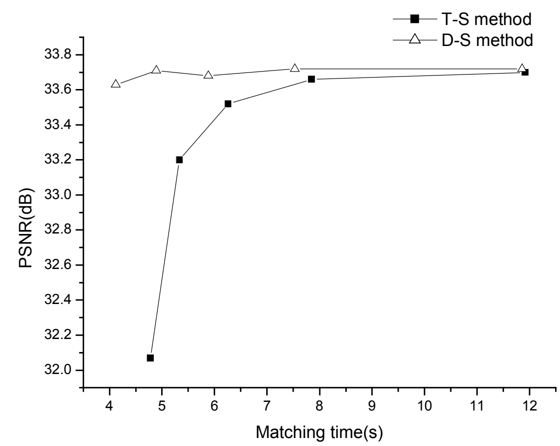

Figure 2. Matching time-PSNR curve of T-S and D-S method

Table 2 gives the result comparison of the above three coding methods in the same compression ratio level. It is clearly shown in the table that P-S method yields better PSNR but poorer matching time and T-S method yields better matching time but poorer PSNR, while D-S method gives best results in both categories.

Similar results can be obtained with different compression ratio level, which shows the method works in both high and low bit rate image coding.

Figure 1 shows that D-S method uses only less than half of the matching time of P-S method to give the same PSNR value. Although it looks nice, it should be noticed that matching time is just a small part of the whole coding time. Maybe some rapid algorithms of preprocessing time are needed for further acceleration of the whole coding process.

Figure 2 gives matching time-PSNR curves of

T-S and D-S method with different match parameter. The five feature points on each curve respectively represent conditions of m=2, m=5, m=10, m=20 and m=50. As is seen from the figure, D-S method is a little faster than T-S method with the same value of m. It could be explained as a result of effectively avoiding much unnecessary regression analysis computation with the use of feature difference decision criterion. An exciting observation here is that D-S method gives a very flat optimal curve. PSNR value is almost the same for m=2 and m=50 but the former case costs only about 1/3 matching time of the latter. It shows that feature difference method can very efficiently find optimal matches almost independent of the value of the match parameter. This property makes it prosperous in applications of other classification algorithms.

Experiments for many other images give similar results. Table 3 lists some typical result data. We can see from the table that D-S method can always get better rate-distortion property in the same level of computational complexity.

5. Conclusion

This paper proposes the notion of feature difference, which is based on the contraction property of transformation in fractal image coding. Then a feature difference decision criterion is detailed and used along with Saupe classification. Experimental results show that the method effectively solves the contradiction of match parameter selection and PSNR value, and generates much higher rebuilt image quality with the same or less coding time. The notion of feature difference is simple but effective. The authors believe that it can be successfully applied to many other classification algorithms.

Reference

[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, Vol 1. No.1, January, 1992, pp18-30

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

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

[5] R. Distasi, M. Polvere, M. Nappi, "Split Decision Functions in Fractal Image Coding", Electronics Letters 34,8, vol. 34, no. 8, pp. 751--753, April 1998.

[6] B. Hurtgen, C. Stiller, "Fast Hierarchical Codebook Search For Fractal Coding of Still Image", EOS/SPIE Visual Communication and PACS for medical applications '93, Berlin 1993.

[7] D. Saupe, "Fractal image compression by multi-dimensional nearest neighbor search", Proceedings DCC'95 Data Compression Conference, J. A. Storer and M. Cohn (eds.), IEEE Comp. Soc. Press, March 1995.

[8] M. Polvere, M. Nappi, "Speed-up Methods in Fractal Image Coding: Comparison of Methods", IEEE Transaction on Image Processing， vol. 9, no. 6, June 2000, pp. 1002-1009.

[9] Arya,S, Mount,D.M, Netanyahu, N.S.,Silverman, R.Wu,A., "An optimal algorithm for approximate nearest neighbour searching", Proc. 5th Annual ACM-SIAM Symposium on Discrete Algorithms (1994) 573-582.

[10] N.Christopher，J.Wein and Ian F.blake, "On the performance of Fractal Compression with Clustering" IEEE TRANS. Image Processing，Vol.5 No.3 March 1996

[11] Y.Fisher, E.W.Jacobs, R.D.Boss, "Fractal image compression using iterated transforms". Technique Report 1408, Naval Ocean Systems Center, San Diago, CA 1991

Table 3. Comparison of T-S and D-S method for different images (m=5)

Image	Coding time		Matching time		Ratio		PSNR	
	T-S	D-S	T-S	D-S	T-S	D-S	T-S	D-S
Lena	14.78	14.29	5.33	4.89	16.14	16.71	33.20	33.71
Barbara	18.29	18.57	8.84	9.06	8.80	8.92	27.94	28.34
Beam	12.42	12.08	2.86	2.58	32.15	33.92	35.19	35.68
Soccer	15.49	14.94	6.04	5.50	14.54	14.71	32.27	32.75