An Efficient Three-Dimensional Fractal Video Coding Method Using Intercube Correlation Search

Chou-Chen Wang
Dept. of Electrical Engineering,
Chung Cheng Institute of Technology, Tashi, Taoyuan.
e-mail: ccwang@cc04.ccit.edu.t

Chin-Hsing Chen
Dept. of Electrical Engineering,
National Cheng Kung University, Tainan.
e-mail:chench@eembox.ncku.edu.t

Chaur-Heh Hsieh
Dept. of Information Engineering,
I-Shou University, Kaohsiung.
e-mail: hsieh@csa500.isu.edu.t

Abstract

An efficient 3-D fractal video coding method that exploits the correlation between range cubes is proposed. Four domain cubes mapped by the previous neighboring range cubes are considered as the good candidate cubes of the input range cube. Simulation results show that when th proposed method is implemented with a fast 3 -D fractal coding algorithm, it can further reduce the encoding times and bit rate with insignificant loss of video quality.

Keywords: *fractal, image compression, video compression, search-order coding.*

1. Introduction

Recently, fractal theor has been widely applied in the field of video compression due to the advantages of resolution independence, fast decompression and high compression ratio [1,2]. However, the major drawback of the fractal video compression is the high encoding complexity to find the best match between a range cube and a large pool of domain cubes. In order to reduce the search complexity, many researchers have presented methods to reduce the search space of the domain pool [2,3,4,5]. They extensivel explored the relationships between the range cube and domain cubes so as to find a method to reduce search space but omitting the correlation between range cubes. It is als noted that in these methods the range cubes within image sequences are processed independently. However, there is usually a high intercube correlation in natural image sequences, thus resulting in a high probability that the neighboring range cubes are matched by a small subset of the domain pool having the similar features or patterns . If one considers the small subset instead of the whole pool of domain cubes, it is possible to further improve the computational efficiency of the encoding process.

There are two major approaches to fractal video coding. The first one is a combination of intra-frame fractal coding and motion compensated inter-frame coding [6,7,8]. This approach is similar to MPEG video coding schemes, using motion compensation techniques coding frames one by one, and taking fractal coding instead of discrete transform coding (DCT). The second one uses an extension of two-dimensional (2-D) image to three-dimensional (3-D) image sequences, having the potential for higher compression than intra coding techniques [3-5]. While both inter/intra and 3-D fractal block coding techniques are resolution independent in the spatial domain, only -D block coding is also resolution independent in the temporal domain. In other words, all three dimensions are contracted in true 3-D fractal coding scheme. In previous publication we propose d the improvement of coding efficiency for fractal coding using intra frame coding [9]. Based on a similar idea of previous proposal [9], we proposed a modified method to be suited to the 3 -D fractal video coding.

Fractal compression of 3-D image sequences can be taken as a direct extension of the techniques involved in coding a 2-D image. Beaumont was the first to extend Jacquin's still image coding technique to video sequence coding [6]. He presented a 3-D fractal-coding scheme using a 12-frame memory, with range cube of 4×4×4 and domain cube of 12×12×12. Although the data compression was impressive, the picture quality was poor and severe blocking artifacts were present. The failure can be mainly attributed to inadequacy of fractal coding in modeling hi gh pass spatial and temporal edge. To further improve the encoding quality, a novel 3 -D partition of input frames is introduced by Lazar and Bruton [3]. However, they suffer from the same drawbacks as 2 -D fractal coding, that is, the computational expense of finding the best matchin between range cube and a domain pool. Therefore, the coding would lag behind capture in a sequence by at a factor of 100, and to achieve a higher quality it may be up to 1000 lower. Clearly, the scheme is not ready for real-ti e applications. Another problem of the Lazar and Bruton's method is to use the spatio-temporal partition to improve decoded quality of high frequency areas. However, the partition needs complex computations and analyses such that the encoding time becomes aggravated.

The key development of this work is to present a more straightforward but efficient method for 3-D fractal video coding, which exploits the correlation between neighborin cubes of the input range cubes and encodes the position and isometry parameters using a search-order code technique in [10]. The proposed method can be applied to any existin 3-D fractal video coding algorithms to further reduce their coding complexity and raise their compression efficiency.

2. Review of The 3D Fractal Vid eo Coding Scheme

Before describing our algorithm, we briefly review the 3-D fractal-coding scheme proposed by Lazar and Bruton [3] in this section. They used 3-D range and domain cubes rather than applied 2-D partition on an intra-frame basis.

Encoding procedure:

The encoding procedure of [3] involves the construction of 3-D range and domain cubes, determining the contraction mapping, and fractal encoding of the parameters. These steps are outlined below

(1) Constructing range cubes and a domain pool
They first partitioned the video sequences into *R-Frames* and *D-Frames*.

c(0, 0)	c(0, 1)	⋯	c(0, n-1)
c(1, 0)	c(1, 1)	⋯	c(1, n-1)
⋮	⋮	⋱	⋮
c(m-1, 0)	c(m-1, 1)	⋯	c(m-1,n-1)

Figure 5. The DCT coefficients for the block of size m×n.

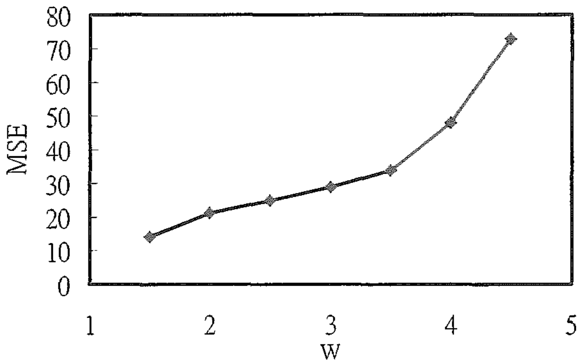
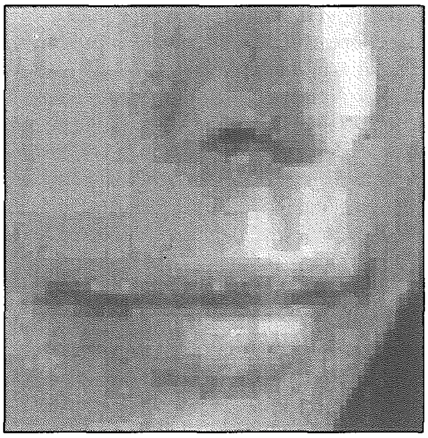
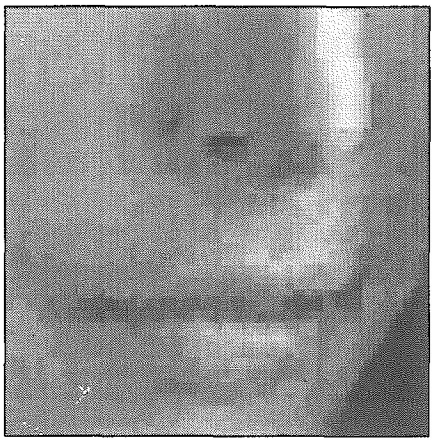


Figure 6. The relationship between MSE and the values of parameter w.



(a) The image encoded by the smooth side-match method.



(b) The image encoded by the side-match method.

(c) The original image.

Figure 7. The reconstructed images at 0.172 bpp.

- The size of nonoverlapping 3-D range blocks (r) and 3-D overlapping domain blocks (d) are set to $B \times B \times T$ and $M_1 B \times M_2 B \times M_3 T$, respectively, where M_1, M_2 are spatial scaling factors, and M_3 is a temporal scalin factor.

- Range cubes are partitioned from *R-Frame*, which are consecutive and nonoverlapping groups of input frames. Each *R-Frame* is restricted to length kT, $k \in Z, k \neq 0$.

- Domain cubes are partitioned from *D-Frame*, which is associated with each *R-Frame*. The length of a *D-Frame* must end at the same temporal location as its associated *R-Frame*, but may start before the beginning of that *R-Frame*. The length is restricted to length lM_3T, $l \in Z, l \neq 0$.

- The *R-Frame* staring at time t is denoted by *R-Frame(t)*, while the associated *D-Frame* is indicated by *D-Frame(t)*. If the parameter values are set to $k = 2$, $l = 4$, $M_1 = M_2 = 2$, $M_3 = 1$, and $T = 4$, the *R-Frame* and its associated *D-Frame* can be illustrated in Figs. 1 and 2.

(2) Domain and range cube mapping:
The fractal approximation of the range cube r_i using affine transformation is given b

$$\tilde{r}_i = \alpha_i I_m(S(d_{N(i)})) + \Delta g \qquad (1)$$

where I_m, α_i and Δg represent the isometry trans-formation, contrast scale factor and luminance shift factor, respectively. $N(i)$ is a domain cube select function, which associates the ith range cube with a domain cube from a domain pool. Typically, the Euclidean metric, $d(\tilde{r}_i, r_i)$, is used to measure th distortion between the range cube r_i and transformed cube $\tilde{r}_i$.

- Spatial scaling function
$S(\bullet)$ is a shrinking operator that averages the size of a 3-D domain block into the same size of 3 -D range block size.

- Isometry operation
For 3-D blocks, considerably more pixel shufflin operations are possible than those for 2-D blocks. To keep the number of isometries to a reasonable value, they restricted the operation of isometries to 8 spatial transformations (intra-frame) and 2 temporal transformations (inter-frame), respectively. Intra-frame basis: pixels within frames are shuffled. Inter-frame basis: the time -reverse ordering of frames themselves. So, $I_m(i) = I_{inter}(i) + I_{intra}(i)$.

- Domain cube search method
In Lazar and Bruton's scheme, a local search for domain cube is conducted, whereby the domain cubes near their corresponding range cube are used. The mapping scheme can be illustrated in Fig.2.

(3) Fractal encoding parameters:
Once the closest domain cube is found, namely, th transformed cube $\tilde{r}_i$ is the best approximation of th given range cube, the parameters and the position of the domain cube are recorded or transmitted.

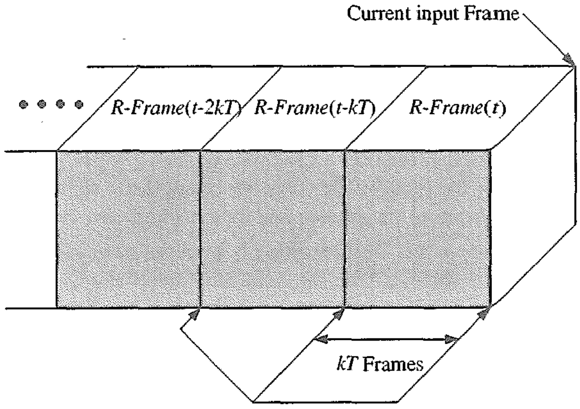


Fig. 1 Partitioning input sequences into *R-Frames* [3].

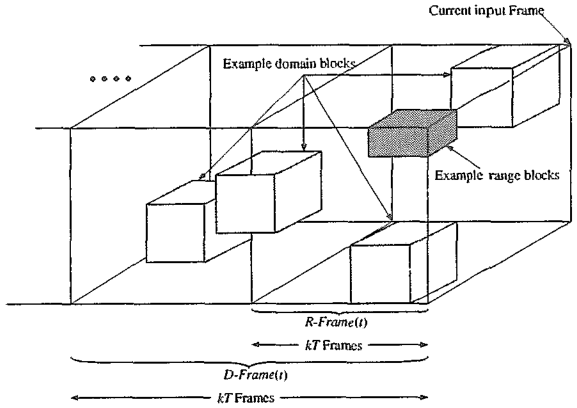


Fig. 2 Example *R-Frame* and associated *D-Frame* [3].

Decoding procedure

The decoding procedure is similar to the 2-D decoding scheme described in [1]. Specially, the domain cubes can be selected from frames in the sequence that has already been decoded. Therefore, the decoder only requires a single iteration to decode. Moreover, the scheme can decode several frames simultaneously.

3. Intercube Correlation Search Algorithm

In order to obtain high quality decoded image sequences, the volume of the range cube us ed in 3-D fractal vide compression should be small. Practically, 4×4×4 and 8×8×4 range cube partitions are the most frequently used in 3-D fractal coding schemes. Therefore, there is usually a high correlation between the neighboring range cubes such that the mapped domain cubes may have highly similar characteristics. In other words, the closest domain cubes of the neighboring cubes may be the good candidates for the input range cube.

Generally, in the 3-D fractal video coding, image sequences are encoded cube by cube in a raster scan order, i.e., from left to right and top to bottom. Our algorithm als searches for the candidate domain cubes in this order. A regular search scheme is employed to search for four causal neighbor cubes in spatial direction, as shown in Fig. 3. The domain cubes, D_1, D_2, D_3 and D_4, represent the four candidate domain cubes, which have been mapped by previous neighboring cubes of the input range cube. The figure also shows the search priority order and the associated 2-b search-order codes. The number '1' represents the first priority and its search-order code is '00', the number '2' the second priority and its search-order code '01', and so on. In this work, we found that the selection of the priority is not critical, because the coding efficiencies are near the same when the different priorities are chosen.

To determine whether a candidate domain cube is good enough for the input range cube, we compute the distortion between the input range cube and the transformed cube $\tilde{r}_i$ of candidate cube. The distortion is represented by mean square error (MSE). After the candidate (one of the D_1, D_2, D_3 and D_4) is found, we check whether it is good enough by comparing its MSE value with a threshold. If it is less than the threshold, the candidate is good enough for the input range cube. Otherwise, it implies that the cube correlation is low and a fast 3-D fractal coding is needed to find the closest domain cube for the input range cube. To demonstrate the efficiency of the proposed algorithm, the fast coding algorithm in [3] is adopted, as an example, in this paper.

There are two search schemes are employed to search for four causal neighbor cubes in the spatial and the spati -temporal directions.

(1) By considering the correlation in spatial domain. The search scheme is employed to search for four domain cubes mapped by causal neighboring cubes of the input range cube in the spatial direction, as shown in Fig. 3.
(2) By considering the correlation both in spatial and temporal domains. The search scheme is employed to search for four domain cubes mapped by causal neighbor cubes of the input range cube in the spatial and temporal directions, as shown in Fig. 4.

The new algorithm can be summarized as follows

(i) Check whether the first candidate domain cube D_1 is 'good'. If it is true, the parameters (search -order code, α_i and Δg) are recorded and the search ends there and proceed to the next range cube in the same manner. Otherwise go to step (ii).
(ii) Repeat step (i) for the next priority neighboring cube until a 'good' candidate domain cube is found or all neighboring cubes are examined. If no good candidate is found, go to step (iii).
(iii) Perform the fast search algorithm proposed by [3] to find the closest domain cube and the parameters (P_D, I_m, α_i, Δg) are recorded, and then go back (i) for the next range cube.

From step (i) and step (iii), it is seen that if a good candidate is found, we send 2 bits search-order code instead of the position P_D and isometry I_m. Therefore, the new algorithm can further reduce the bit rate in addition t computational complexity. In the decoder, the image cubes are also reconstructed in a raster scan order. The search procedure is the same as that of the encoder. To let the decoder distinguish parameters (P_D, I_m) from search-order code, an extra indicator bit is needed in transmission. According to the search procedure and the search -order code received, we can easily recover the parameters P_D and I_m of the input cube just using the two parameters represented in the previous cube by the search -order code. Although the first search scheme only use s the spatial correlation, it doesn't require extra memory to store the previous decoded image frames. The second method considers the temporal correlation except utilizing the spatial correlation.

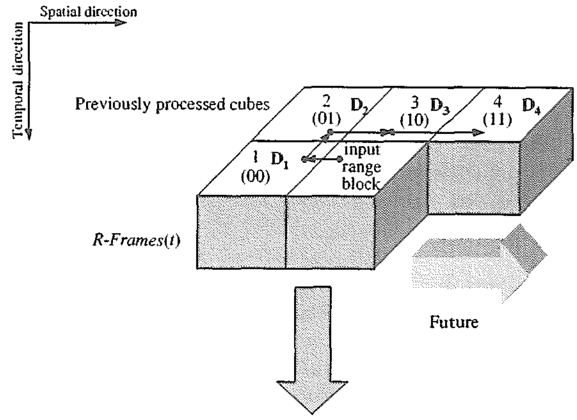


Fig.3 Intercube correlation search scheme in spatial direction.

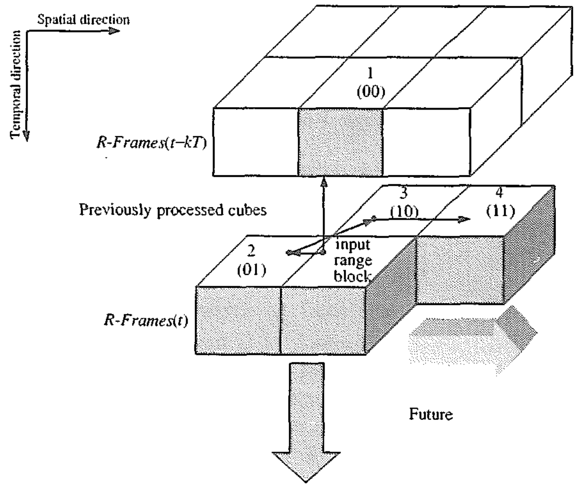


Fig. 4 Intercube correlation search scheme in spatial and temporal directions.

Some researchers have conducted experiments and proved the advantages of fractal video coding includin resolution independence and high compression ratio. Fisher *et al.* have demonstrated decoding video sequences at higher spatial resolutions [7], while Barakat *et al* [5] have simulated decoding at higher spatial and temporal resolutions using 3-D block coding with satisfactory results.

In Lazar and Bruton's experiments, however, they onl performed a spatial contraction but left the time scale of the domain cubes unchanged. This will lead to the unsuccess of temporal resolution independence at decoding image sequences. It has been shown that if using a co ntraction operator with spatial and temporal contraction, the motion of objects can be described with more accuracy [4]. Therefore, in this work, we use contraction in all three dimensions. It is noted that no motion compensation is necessary to describe temporal redundancies, as the 3-D domain cube contains the motion information. Therefore, the length of *D-frames* is larger than that of *R-frames* for our experiments in this chapter, that is, $lM_3 > k$.

4. Experimental Results

The CCITT test sequences "Miss Am erica", which is used in the development of video phone/video conference standard, was used to test the experimental results in this work. The picture size is 352×288 pixels and grayscale is

256 levels. The input *R-frame* (kT) is first partitioned into $B \times B \times T$, and each associated with *D-frame* (lM_3T) is partitioned into $M_1B \times M_2B \times M_3T$.

In order to exhibit the successes of our methods, a local search algorithm similar to that of Lazar and Bruton's [3 without using spatio-temporal partition is also implemented. The range of the domain cubes searched is ±8 for spatial direction (x, y) and −4 for temporal direction (t), due to the causality of video sequence, with 1 pixel step size. In other words, the number of domain cubes compared by each range cube is 16 ×16×4 = 1024 cubes. The position of the closest domain cube mapped by the input range cube is indicated by $P_D = (n_1+x, n_2+y, n_3+t)$, where the (n_1, n_2, n_3) is the address of the input range cube.

Figure 5 shows the average rate-distortion curve for our search scheme based on the local search in [3]. It indicates that the quality of the reconstructed frames start to degrade noticeably when $t > 16$. Similar to previous work [9], w take $t = 16$ to test all results in this paper. For the purpose of comparison, the parameters α_i, Δg, and I_m were uniform quantized using 5, 7 and 4 bits, respectively. For the isometries used herein include 8 intra -frames and 2 inter-frame isometries described in Sec. 2. One indicator bit is also needed in transmission to let the decoder distingui sh parameters (P_D, I_m) from search-order codes. The compression ratio (CR) of the proposed algorithm is calculated b

$$CR = \frac{W \times H \times T \times 8}{N_I \times (2 + L_b) + (N_r - N_I) \times P_b + N_r \times 1} \quad bpp \qquad (2)$$

where

N_I = total number of range cubes using the intercube cor-relation search.

L_b = total number of bits to encode the parameter (α_i, Δg).

N_r = total number of range cubes.

P_b = total number of bits to encode the parameter (P_D, I_m, α_i, Δg).

$W \times H \times T \times 8$ = total number of bits to represent 3-D image sequences.

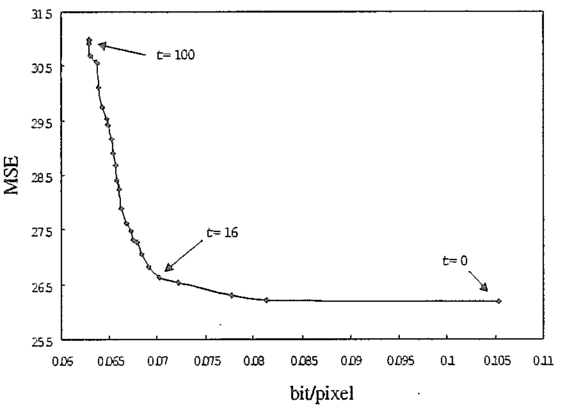


Fig. 5 The rate-distortion curve.

The improvements are to be expected when the 3 -D fractal coding is applied to the background and the slow motion object of the video sequences. But, the quality in the region with large motion degrades distinctly, since the temporal correlation between 3-D range blocks is reduced in the region. Just like any 3-D coding schemes, e.g., 3-D discrete cosine transform, subband coding, etc., 3 -D fractal coding is not able to handle a large amount of frame motion This is because larger inter -frame motion will destroy the self-transformation in image volumes. In other words, the correlation exiting the temporal domain is very low. In order to preserve th e image quality of the region with a motion, we encode a group of four frames each time, namely, the length of *R-frame* for our experiments is set t $T = 4$. Also, to fulfill the contractive requirement in the time domain, the length of *D-frame* is set to $M_3T = 8$.

The intercube correlation search schemes, shown in Figs. 3 and 4, based on [3] are conducted in this section. The compression ratio (CR) and PSNR for different search methods are shown in Fig. 6. Although very high the compression ratio can be obtained when the spatial range size is increased, the blocking artifacts can be clearly found. This is because the compression ratio and image quality is a trade-off. Table 1 compares the average encoding time, PSNR, and CR. The results indicate that the prop osed methods can further speed up the encoding process of local search [3] over a factor of 8. The average compression ratios are 31 and 114 for 4×4×4 and 8×8×4 range partitions, respectively. Thus, the CR raises are (31 −19.6) / 19.6 × 100% = 58.2% and (114.1−78.8) / 78.8 × 100% = 44.8%, respectively. The performance improvement of 4 ×4×4 range partition is greater than that of 8×8×4 range partition due to the higher intercube correlation in 4 ×4×4 range partition and the more cubes performed using our method. The loss of image quality is insignificant because all the differences in PSNR obtained from [3] and our method are less than 0.2 dB. We also can find the performance of the time-spatial correlation search scheme is slightly better than that of spatial c orrelation search scheme. However, an extra memory is required to store the previous decoded frames. It is impractical to realization the video coding scheme.

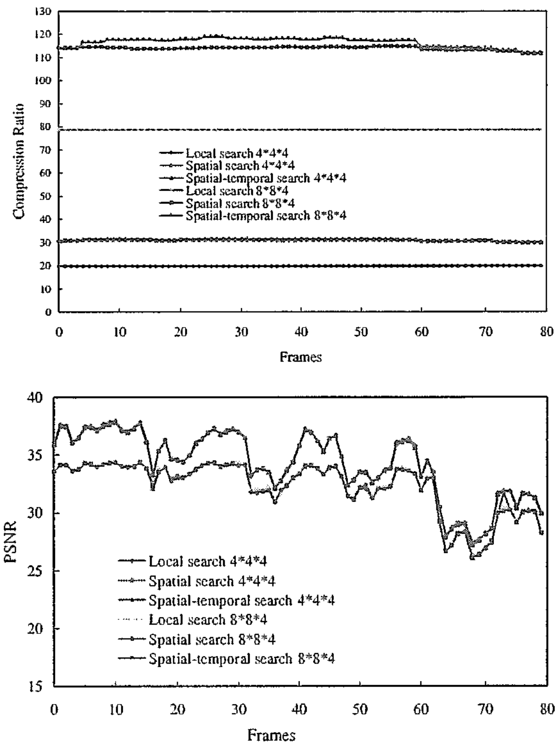


Fig. 6 Compression ratio and PSNR for "Miss America" sequence.

5. Conclusions

In this paper, we present an efficient search strategy to speed up the encoding process of the 3-D fractal video coding. In addition, a search-order coding technique developed in [10] is used to further raise compression ratios Two search methods, which take advantage o f the correlation between intercubes in the spatial and temporal directions, are presented. The results indicate that average compression ratios of 31 to 114, and a speedup factor of 8 with a slight loss of quality, as compared to the local search method. It can be performed in real-time and the subjective quality is suitable for video-conferencing applications. Further work will be devoted to developing low-complexit 3-D fractal coding algorithm to suitable for very low bit rate video coding systems.

Acknowledgments

This work is supported in part now by the National Science Council, Taiwan, R.O.C. under grant NSC 89-2213-E-014-021.

References

[1] Y. Fisher, *Fractal image compression - Theory and Applications*. Springer-Verlag, New York, 1994.
[2] T. C. Ferguson and H. R. Wu, "Fractal transfor techniques for very low bit rate video coding," *IEEE Int. Symposium on Circuits and Systems* , pp 1456-1459, June 1997.
[3] M. S. Lazar and L. T. Burton, "Fractal block coding of digital video," *IEEE Trans. on CASVT* , vol. 4, pp. 297-308, June 1994.
[4] K. U. Barthel and T. Voye, "Three-dimensional fractal video coding," *In Proc. of ICIP'95*, vol.3, pp. 260-263, Washington D.C., Oct. 1995.
[5] M. Barakat and J. L. Dugelay, "Image sequence coding using 3-D I.F.S.," *In Proc. of ICIP'96*, pp.1 1-144, Lausanne, Sept. 1996.
[6] J. M. Beaumont, "Image data compression usin fractal techniques," *BT Techn. J.*, vol.9, no. 4, pp. 93-109, 1991.
[7] Y. Fisher, D. Rogovin and T. P. Shen, "Fractal (self-VQ) encoding of video sequences," In *SPIE Proc., Visual Communication and Image Processing* , Chicago, Sept. 1994.
[8] X. Dong and R. Sudhakar, "Fractal compression of image sequences," *In Proc. of ICIP'95*, vol.3, pp. 284-287, Washington D.C., Oct. 1995.
[9] C. C. Wang, C. H. Chen and C. H. Hsieh, "Fast fractal image coding using interblock correlation," *The 12th Computer Vision, Graphics, and Image Processing Symposium (CVGIP'99)*, pp. 443-448, Wanli, Taiwan, Aug. 1999.
[10] C. H. Hsieh and J. C. Tsai, "Lossless compression of VQ index with search-order coding," *IEEE Trans. on Image Processing*, vol. 5, pp. 1579-1582, Nov. 1996.

Table 1 Comparison of coding results using local search, spatial correlation search and spatial-temporal correlation search, respectively.

Range size	Methods	Time (sec)	PSNR (dB)	CR
4×4×4	Local search	181	35.72	19.6
	Spatial search	24	35.53	30.8
	Spatial-temporal search	23	35.57	31.0
8×8×4	Local search	163	33.91	78.8
	Spatial search	39	33.65	114.1
	Spatial-temporal search	37	33.71	116.5