

RESEARCH

Open Access

A novel fractal monocular and stereo video codec with object-based functionality

Shiping Zhu*, Liyun Li and Zaikuo Wang

Abstract

Based on the classical fractal video compression method, an improved monocular fractal compression method is proposed which includes using more effective macroblock partition scheme instead of classical quadtree partition scheme; using improved fast motion estimation to increase the calculation speed; using homo-I-frame like in H.264, etc. The monocular codec uses the motion compensated prediction (MCP) structure. And stereo fractal video coding is proposed which matches the macroblock with two reference frames in left and right views, and it results in increasing compression ratio and reducing bit rate/bandwidth when transmitting compressed video data. The stereo codec combines MCP and disparity compensated prediction. And a new method of object-based fractal video coding is proposed in which each object can be encoded and decoded independently with higher compression ratio and speed and less bit rate/bandwidth when transmitting compressed stereo video data greatly. Experimental results indicate that the proposed monocular method can raise compression ratio 3.6 to 7.5 times, speed up compression time 5.3 to 22.3 times, and improve the image quality 3.81 to 9.24 dB in comparison with circular prediction mapping and non-contractive interframe mapping. The PSNR of the proposed stereo video coding is about 0.17 dB higher than that of the proposed monocular video coding, and 0.69 dB higher than that of JMVC 4.0 on average. Comparing with the bit rate resulted by the proposed monocular video coding and JMVC 4.0, the proposed stereo video coding achieves, on average, 2.53 and 21.14 Kbps bit rate saving, respectively. The proposed object-based fractal monocular and stereo video coding methods are simple and effective, and they make the applications of fractal monocular and stereo video coding more flexible and practicable.

Keywords: Fractal, Monocular, Stereo, Video coding, Object-based

Introduction

There are several image/video compression methods, for example, JPEG, MPEG, and H.26X which are all based on motion estimation/compensation (ME/MC). But fractal compression, which is based on the iterated function system (IFS) proposed by Mandelbrot [1], is a relative new approach to image coding. It reduces the redundancy of images by using their self-similarity properties which can make high compression ratio and simplicity of decompression. So, if we want to store a picture, we can do it by storing the numbers that define

the contraction maps [2] and simply generating the picture whenever we want to see it. And the image can be decoded in any scale. Hence, fractal compression is particularly suitable for the situation of one encoding and many decoding. But some images do not contain the type of self-similarity, such as an image of a face. Fortunately, the human eyes are insensitive to a wide variety of information loss, so we allow some errors in our representation of the image of the face as a set of self-transformations. However, it usually takes long time to encode. In order to speed up the fractal encoder, Lin and Ming-Sheng [3] proposed an edge property-based neighborhood region search method. But the image quality is decaying. In this article, we proposed a novel macroblock partition scheme combined block searching

* Correspondence: spzhu@163.com

Department of Measurement Control and Information Technology, School of Instrumentation Science and Optoelectronics Engineering, Beihang University, XueYuan Road No. 37, HaiDian District, Beijing 100191, China

method in fractal coding to obtain optimal performance.

With regard to video compression, two basic fractal compression methods are used most frequently. One is known as the cube-based compression [4-6] and the other is frame-based [7,8]. In the cube-based compression, a sequence of images is divided into groups of frames, each of which in turn is partitioned into non-overlapped cubes. It can obtain high-quality decompressed images while has a high computing complexity and low compression ratio. In the frame-based compression, the domain blocks from the previous frame are used to compute the approximate transformation for the range blocks of the current frame. Although it obtains a high compression ratio, the current frame is related to the previous frame which introduces and spreads errors between frames. In this article, the two methods are combined which is also researched in [9-11] in order to improve the results instead of choosing the optimal one. In addition, ME is one of the most time-consuming parts in video coding. It is important to develop fast and effective ME algorithms. A novel fast ME method is proposed which performs a "rough" search before a "precise" search for the best partition in fractal coding. By reducing the searching load for the non-best partitions, the computation complexity for search can greatly be decreased.

In general, stereo video sequences are composed of left and right images acquired from two slightly different viewpoints, thus making them similar and containing a lot of redundant information. Fractal compression is an effective method to remove the redundancy. But traditional (2D) fractal coding makes the depth perception defective. In this article, we proposed disparity compensated prediction (DCP) and motion compensated prediction (MCP) which are used in fractal stereo coding to conquer these problems.

In this article, object-based (OB) coding, the notion which is first used by the MPEG-4 standard [12], is researched in fractal video coding. We developed a novel OB-video coding algorithm which has important advantages: it allows manipulation of image objects without complete decoding of the stream, and then improves the coding quality and reduces the bit rate. It alleviates the problem of annoying coding effects, such as blocking artifacts and mosquito effects compared to block-based approach at low bit rate, especially when the blocks coincide with boundaries of different objects. The object-based approach can also

provide more natural representations of the scene and has another potential benefit of acquiring the depth information of semantically meaningful objects. In such a scheme, a *prior* segmentation map (*alpha* plane) of the image, which segments the image into objects, is known in advance [13,14].

The rest of the article is organized as follows. The theory of fractal coding is summarized in Section 2. The proposed improving methods for monocular fractal video sequence coding are presented in Section 3. The method of stereo fractal video compression and decompression is proposed in Section 4. A detailed design of a new object-based fractal compression of monocular video sequence is presented in Section 5. The experimental results are presented in Section 6. And finally the conclusions are outlined in Section 7.

The fractal compression mathematical theory

Let $I(X)$ be image intensity of a pixel at position $X = (x, y)$ and let $\{R_1, \dots, R_N\}$ be the set of N non-overlapping range blocks (i.e., collections of pixel coordinates) partitioning the image. Similarly, let $\{D_1, \dots, D_M\}$ be the set of M , possibly overlapping, domain blocks covering the image. Finally, let $I_{R_i} = \{I(X) : X \in R_i\}$ and $I_{D_j} = \{I(X) : X \in D_j\}$.

In general, the size of a range block, denoted as $n \times m$, could have n and m chosen as 16, 8, or 4. For each range block $R_i (i = 1 \dots N)$, the goal is to find a domain block $D_j (j = 1 \dots M)$ and a contractive mapping w_i that jointly minimize a dissimilarity (distortion) criterion ε . The contractive affine mapping w_i consists of three submappings.

(1) *Contraction* $\sigma(I, X)$: The dimension of R_i is $m \times n$, which is not the same as the dimension $2m \times 2n$ of D_j , so they cannot be compared. The function $\sigma(I, X)$ is to shrink the domain block D_j by averaging the intensities of its four neighboring pixels $[(I^k, (k = 1 \dots 4))]$ of disjoint groups leading to the same dimension block denoted symbolically as D_j^r , which is also known as the codebook block expressing with the dimension $m \times n$. If the intensity of D_j is expressed as the submatrix

$$I_{D_j}(x_1, y_1), \quad 1 \leq x_1 \leq 2n, \quad 1 \leq y_1 \leq 2m$$

and D_j^r is expressed as the submatrix,

$$I_{D_j^r}(x_2, y_2), \quad 1 \leq x_2 \leq n, \quad 1 \leq y_2 \leq m$$

$$\sigma(I, X) : I_{D_j^r}(x_2, y_2) \xleftarrow{\sigma(I, X)} I_{D_j}(x_1, y_1)$$

That is

$$I_{D_j^r}(x_2, y_2) = \frac{I_{D_j}(2x_2, 2y_2) + I_{D_j}(2x_2 + 1, 2y_2) + I_{D_j}(2x_2, 2y_2 + 1) + I_{D_j}(2x_2 + 1, 2y_2 + 1)}{4} \quad (1)$$

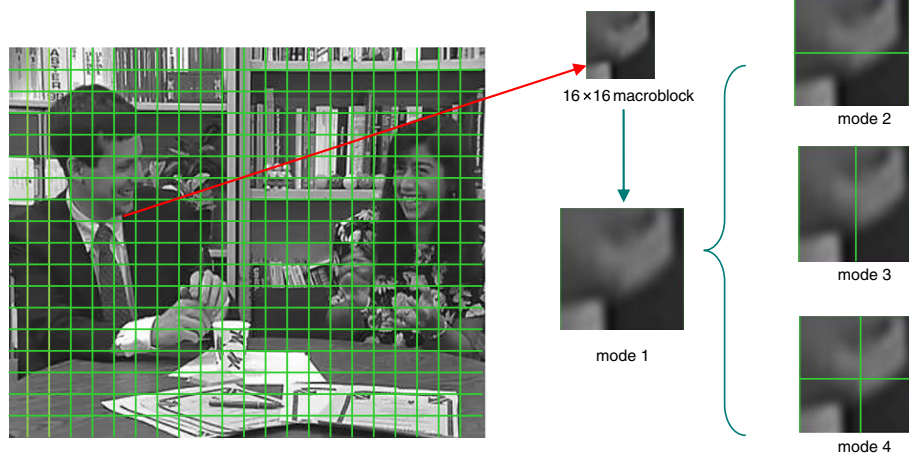


Figure 1 Microblock partition mode.

(2) *Geometric transformation*:

$$\xi(X) \triangleq AX + b \quad (2)$$

where $X = (x, y)$, A is a 2×2 matrix and b is a translation vector (this mapping must be 1-to-1 between pixels of the range and codebook block). The above general expression can be simplified by constraining the transformation A to eight cases: four rotations ($0^\circ, 90^\circ, -90^\circ, 180^\circ$) and four mirror reflections (mid-horizontal, mid-vertical, first diagonal, and second diagonal) [15,16]. $\{\zeta_i^p\}_{p=1}^8$ denotes the set of possible transformations A .

(3) *Photometric transformation*: We define $\gamma \odot I(X)$ as the following to adjust grey level:

$$\gamma \odot I(X) \triangleq s \cdot I(X) + o \quad (3)$$

where \odot is the composition operator, s is a scaling factor which controls the contrast, and o is an offset which controls the brightness of the transformation. The above general expression accounts for different dynamic ranges of pixels in the range and domain blocks.

The overall transformation w_i that maps a domain-block pixel into the range-block pixel at X is

$$w_i(I_{D_j}, X) \triangleq \gamma \odot \sigma(I_{D_j}, \zeta_i^p(X)), X \in D_j, p \in \{1, \dots, 8\} \quad (4)$$

We also can write w_i as the following:

$$w_i(I_{D_j}, X) \triangleq s_i \cdot \sigma(I_{D_j}, \zeta_i^p(X)) + o_i, X \in D_j, p \in \{1, \dots, 8\} \quad (5)$$

In order to encode range block R_i , a search for index j (domain block D_j) and for an isometry ζ_i^p must be executed, jointly with the computation of photometric parameters s_i and o_i . This can be performed by minimizing

the following mean-squared error

$$\varepsilon(I_{R_i}, I_{D_j}, w_i) = \frac{1}{|R_i|} \sum_{X \in R_i} [I_{R_i}(X) - w_i(I_{D_j}, X)]^2 \quad (6)$$

where $|R_i| = \text{Card}(R_i)$, which can get the number of pixels that R_i contains. While the isometry ζ_i^p and index j (equivalent to translation b) are usually found by exhaustive search, the scaling s_i and offset o_i are computed as follows

$$s_i = \frac{\sum_{X \in R_i} [\sigma(I_{D_j}, \zeta_i^p(X)) - m_{D_j}] [I_{R_i}(X) - m_{R_i}]}{\sum_{X \in R_i} [\sigma(I_{D_j}, \zeta_i^p(X)) - m_{D_j}]^2} \quad (7)$$

$$o_i = m_{R_i} - s_i \cdot m_{D_j} \quad (8)$$

where m_{R_i} and m_{D_j} are the mean intensity values in the range and domain blocks, respectively. Equations (7) and (8) will give us contrast and brightness settings that make the affinely transformed $I_{R_i}(X)$ values have the least

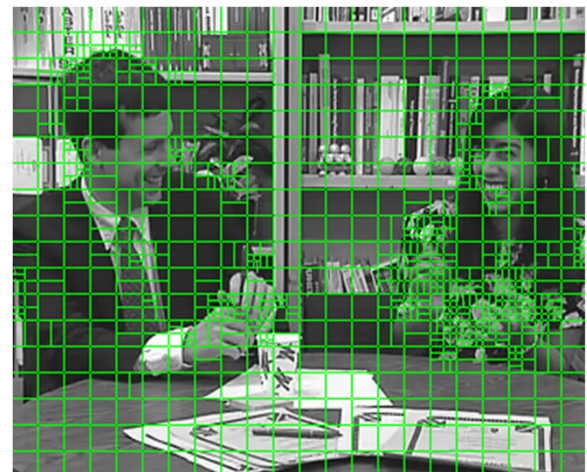
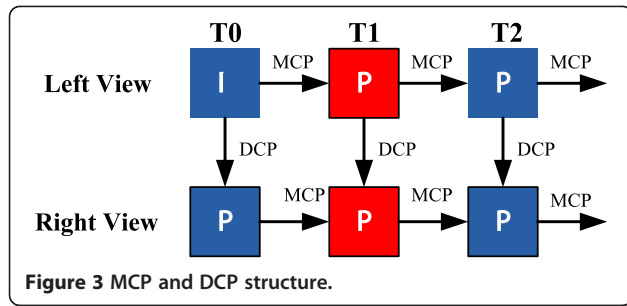


Figure 2 Result of microblock partition.



squared distance from the $w_i(I_{D_j}, X)$ values. This permits a precise representation of the mean local intensity but to assure convergence at the decoder requiring a modification of the photometric transformation, without a constraint on the intensity scaling coefficients [17]. This can be considered as orthogonalization with respect to the constant blocks and has been treated in detail in [18]. So, the matching rule in fractal image coding is RMS:

$$RMS = \frac{1}{N} \left[\sum_{i=1}^N r_i^2 + s \left(s \sum_{i=1}^N d_i^2 - 2 \sum_{i=1}^N r_i d_i + 2o \sum_{i=1}^N d_i^2 \right) + o \left(N \cdot o - 2 \sum_{i=1}^N r_i \right) \right] \quad (9)$$

$$s = \frac{N \sum_{i=1}^N r_i d_i - \sum_{i=1}^N r_i \sum_{i=1}^N d_i}{N \sum_{i=1}^N d_i^2 - \left(\sum_{i=1}^N d_i \right)^2} \quad (10)$$

$$o = \frac{1}{N} \left[\sum_{i=1}^N r_i - s \sum_{i=1}^N d_i \right] \quad (11)$$

where r_i, d_i ($i = 1 \dots N$) are pixel value of range block (R) and domain block (D) and N is the number of pixels in the block.

A new fractal monocular video coding method

Macroblock partition

Macroblock partition has a large impact on calculation speed and complexity of video compression algorithm. In circular prediction mapping and non-contractive inter-frame mapping (CPM/NCIM) [19], a frame is partitioned by quadtree partition and the iteration is used in matching

process, resulting in high calculation complexity. In this article, macroblock partition scheme like in H.264 is used which reduces the number of the blocks compared to the quadtree partition. A frame is partitioned into many fixed size (generally 16×16 pixels) macroblocks, and then each macroblock may be partitioned in four ways and motion compensated either as one 16×16 macroblock partition, two 16×8 partitions, two 8×16 partitions or four 8×8 partitions as shown in Figure 1.

Before the block matching processing, RMS of the whole microblock is calculated in mode 1, and γ is defined as a threshold. Encodings made with lower γ will have better fidelity, longer encoding time, and those with higher γ will have worse fidelity, shorter time. In general, we let

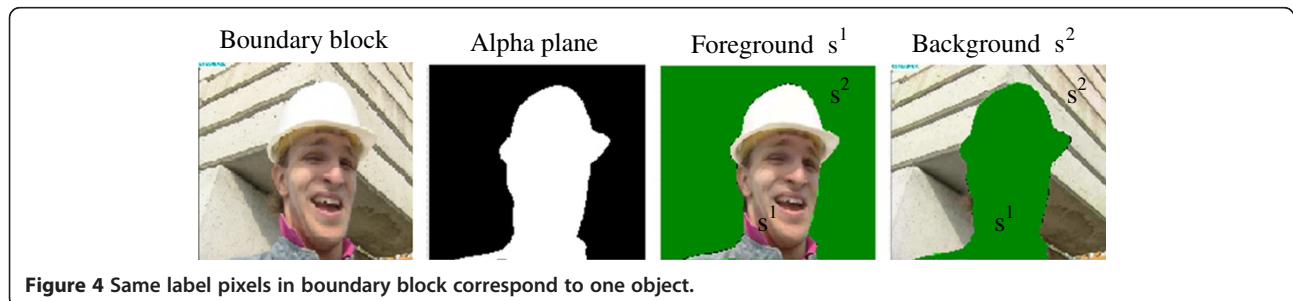
$$\gamma = t \times t \times no$$

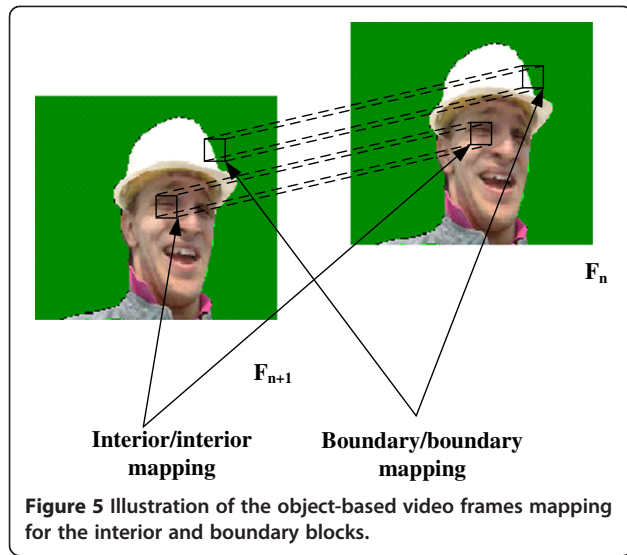
t is dependent on the size of the range block. With enormous practice, when the range block is 16×16 , t is 10.0; when the range block is 8×8 , t is 8.0; when the range block is 4×4 , t is 6.0. Thus, we can get good performance. no is the pixel number of the range block. The steps of macroblock partition are as follows.

First, RMS which is calculated in mode 1 is compared to γ . If RMS is less than γ , then current IFS is saved and the algorithm processes the next block matching. Otherwise RMS will be calculated after the whole microblock is partitioned in mode 2. If RMS is more than γ , then mode 3 will be used. However, if RMS in mode 3 is also more than γ , mode 4 will be used automatically. Taking into account that the RMS of the four 8×8 blocks in mode 4 is more than γ , the block can be partitioned in a further four ways either as one 8×8 sub-macroblock partition, two 4×8 partitions, two 8×4 partitions, or four 4×4 partitions to find the matching block in the same way. And the result of microblock partition is shown in Figure 2. In areas where there is little change between the frames (RMS is little), a 16×16 partition is chosen; in the areas of detailed motion, smaller partitions are more efficient.

A fast ME method

The most important factors which affect the fractal compression ratio and speed are the number of the domain blocks which need be searching and the ME method.





The first one is discussed in Section 3.1. ME is time-consuming in video coding. We proposed a fast ME method which reduced the block searching strategy and range to increase the calculation speed greatly. Since there are temporal and spatial relations between two frames, the mapping block (domain block) is often near by the corresponding location in the reference frame of the range block. And the searching range is limited from 7 to 15 pixels from the corresponding location. So, the calculation complexity is decreased.

Using homo-I-frame in H.264

The original reference frame (homo-I-frame in H.264) makes a great impact on compression ratio and decoding image quality. In CPM/NCIM, the original reference frames are coded by using CPM, and the original reference frames could be several frames.

But in CPM, the coding process involves complex block-classifying, block-overturning, and iteration in order to make decoding frames converge to original frames, so the compression performances are under the requirements. Then, the method based on the discrete cosine transform (DCT) which is expressed in Equation (12) and has worked effectively in JPEG image compression standard is used to treat the original reference frame [20].

Let X_{ij} denote a pixel, $i = 0 \dots N, j = 0 \dots N$, with an $N \times N$ block. The Y_{xy} is an $N \times N$ matrix which stores DCT coefficients. Taking the DCT used in JPEG,

$$Y_{xy} = \frac{1}{4} C_{xy} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{ij} \cos \frac{(2j+1)y\pi}{2N} \cos \frac{(2i+1)x\pi}{2N} \quad (12)$$

$$\text{with } C_x = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for } x = 0 \\ 1, & \text{for } x > 0 \end{cases} \quad C_y = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for } y = 0 \\ 1, & \text{for } y > 0 \end{cases}.$$

And the inverse DCT (IDCT) which is used in decoder

$$X_{ij} = \frac{1}{4} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C_x C_y Y_{xy} \cos \frac{(2i+1)x\pi}{2N} \cos \frac{(2j+1)y\pi}{2N} \quad (13)$$

Stereo fractal video coding and decoding

Stereo fractal video coding

The most mature technique for the multi-view video sequences compression is the method defined in the MPEG-4 multi-view profile [21]. With this approach, for example, the coder first compresses the right view with a monoscopic video sequence coding algorithm. To code the left view, each macroblock is predicted both from the right view using DCP, and from the previous frame of the right view using MCP as shown in Figure 3. The predictive residual is then coded using the one which gives smaller predictive residual.

So, in this article, we make use of this method based on fractal video compression algorithm which is presented in Section 3. For right view frames, the coder searches the D block in the previous frame of right view using domain block searching strategy and range in Section 3.2, but for left view frames, the coder searches the D block both in the right view frame (DCP) and left previous frame (MCP), and in two D blocks, the block which has smaller RMS is the best matching.

Stereo fractal video decoding

We cannot decode the right and left view videos simultaneously since the right view video which uses DCP to

Table 1 Comparison of coding results of four video sequences

Sequences	PSNR (dB)			Compression ratio			Compression time (s)		
	CPM/NCIM	Proposed	H.264	CPM/NCIM	Proposed	H.264	CPM/NCIM	Proposed	H.264
Hall	26.25	35.49	37.98	13.22	98.35	113.30	15.36	0.88	1.91
Highway	31.72	35.53	38.89	22.60	81.18	105.84	9.80	0.94	1.89
Race	25.07	32.60	38.69	10.82	38.53	36.22	20.93	3.95	6.11
Bridge-close	26.06	32.40	35.75	9.30	63.97	55.62	22.52	1.01	2.01

Table 2 Experimental results comparison between the proposed method with [22,23]

Sequences 256 × 256 pixels	Performance	Methods		
		Proposed	Reference [22]	Reference [23]
Videoconference	Compression ratio	83.91	11.76	16.02
	PSNR (dB)	34.67	33.79	35.40
	Total encoding time (min)	0.23	20.00	18.05

encode cannot decode independently. For example, we decode the third frame of the right video after decoding the reference frame of the left video. The difference between using CPM/NCIM and using homo-I-frame in fractal decomposition is that the decompression method is IDCT transformation in the latter method which using DCT transformation to encode. Fractal decompression is an iterative process which uses the following equation:

$$R_i \cong s_i \cdot D_{a(i)} + (o_i - s_i \cdot d) \cdot C \quad (14)$$

where d and C are the DC part of the domain block ($D_{a(i)}$) and the coefficients of the DCT transformation, respectively. R_i denotes range block. s_i and o_i are known as the scaling and the offset factors.

Object-based fractal video coding

Object-based fractal video compression is proposed in this article. The objects can be defined by a *prior* segmentation map named *alpha* plane and are encoded independently of each other.

The details of the proposed method are as following: the R and D blocks remain rectangular. If all pixels of the R block are inside current coded object, the R block is called interior block. If some pixels of R block are inside current encoded object but some are outside, it is called boundary block. The position of R and D block must be same to each other when searching and matching, interior block matching interior block, boundary block matching boundary block. Coding interior block is the same as NOB, so the key of OB is how to code boundary block. It is obvious that they contain pixels from two or more objects (i.e., the foreground and background). Therefore, in order not to mix pixels from different objects within one transformation, we associate the *alpha* plane a label with each pixel. It means that same label pixels are from one object as shown in Figure 4.

It is supposed that current coded object is object 1 as S^1 , so the method of coding boundary block is as following: for the relative computing of R block, only pixels inside S^1 are calculated, pixels inside S^2 are not considered; for the relative computing of D block, as is shown in Equation (15),

If the pixel d_i within D block corresponds to the same position of R block belongs to S^1 , then its original value ($I(d_i)$) is used, otherwise the average value of pixels (\bar{I}) inside D block which belongs to S^1 is assigned to d_i , resulting the new D block intensity $I_{r_i}(d_i)$. Illustration of the object-based video frames mapping for the interior and boundary blocks is shown in Figure 5.

Experimental results

Monocular video coding

To evaluate the performance of the proposed monocular codec, we use four video sequences: "hall.cif" (352 × 288 pixels, 15 frames), "highway.cif" (352 × 288 pixels, 15 frames), "race.yuv" (640 × 480, 15 frames), and "bridge-close.cif" (352 × 288 pixels, 15 frames). The maximum and minimum partition block sizes are 16 × 16 pixels and 4 × 4 pixels, respectively. To compare the performances with other methods, H.264 (main profile, JM 15.1, Search range: 7 pixels, Type of block matching algorithm: UMHexagon Search, QP: 28, Adopted fractional-pixel accuracy: 1/4 pixel, Entropy coding method: CAVLC) and CPM/NCIM are used. The experiments are proceeded in a PC (OS: Microsoft Windows XP Professional, CPU: Inter® Pentium® D, 3.20 GHz, RAM: 2048 MB).

The comparison of average coding results of four video sequences is shown in Table 1. The results indicate that the proposed method can raise compression ratio 3.6 to 7.5 times, speed up compression time 5.3 to 22.3 times, and improve the image quality 3.81 to 9.24 dB in comparison with CPM/NCIM. Although the values of PSNR are lower than H.264, they are all above 32 dB and the human eyes are insensitive to the differences. The compression ratios of the proposed method are near to H.264, and some are higher than that of H. 264, while the compression speed is much better than H.264, which speed up compression time 1.93 times on average. So, the proposed method leads to more real-time applications.

The comparison is shown in Figure 6 for 15 frames of "bridge-close.yuv". It is obvious that the proposed method has better performances than CPM/NCIM, and also has advantages compared to H.264. Besides, the

$$I_{r_i}(d_i) = \begin{cases} I(d_i), & \text{if } d_i \in S^1 \text{ and the corresponding position in } R, r_i \in S^1 \\ \bar{I}, & \text{if } d_i \in S^1 \text{ and the corresponding position in } R, r_i \notin S^1 \end{cases} \quad (15)$$

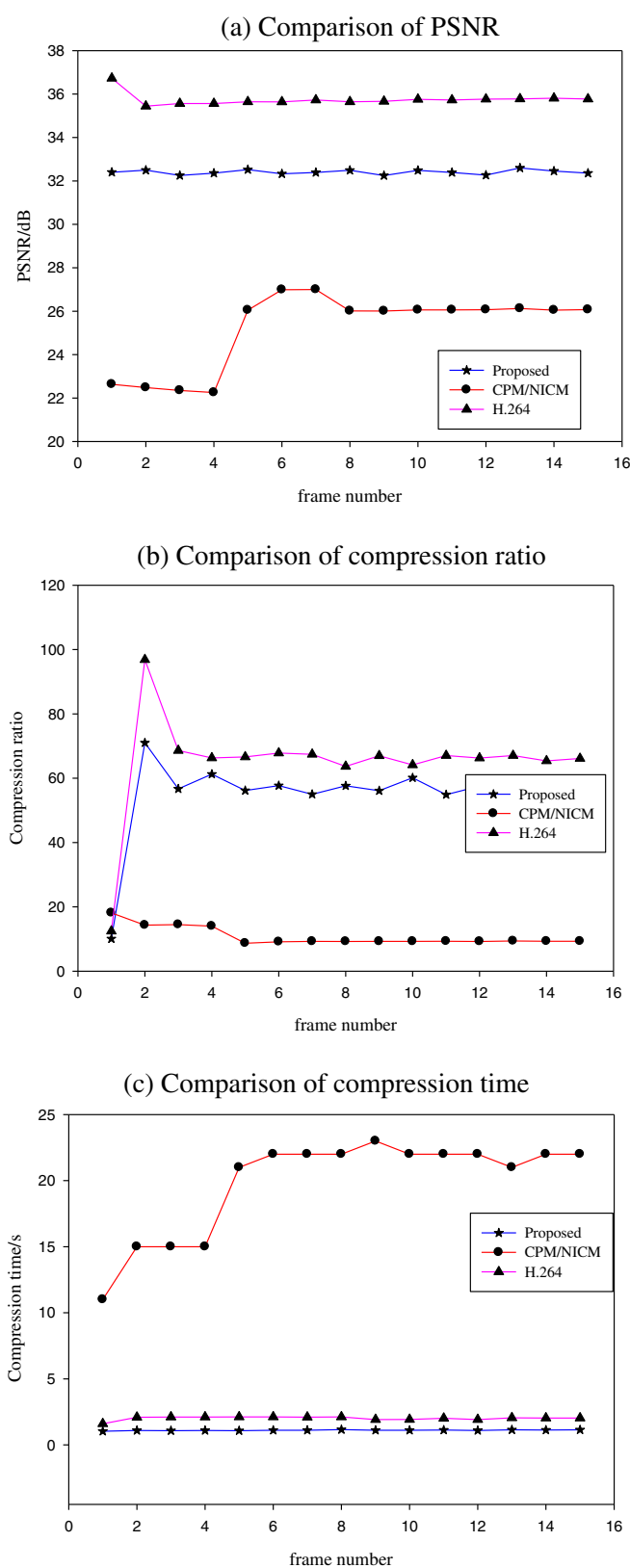


Figure 6 The experimental comparison of "bridge-close.yuv". (a) Comparison of PSNR. (b) Comparison of compression ratio. (c) Comparison of compression time.

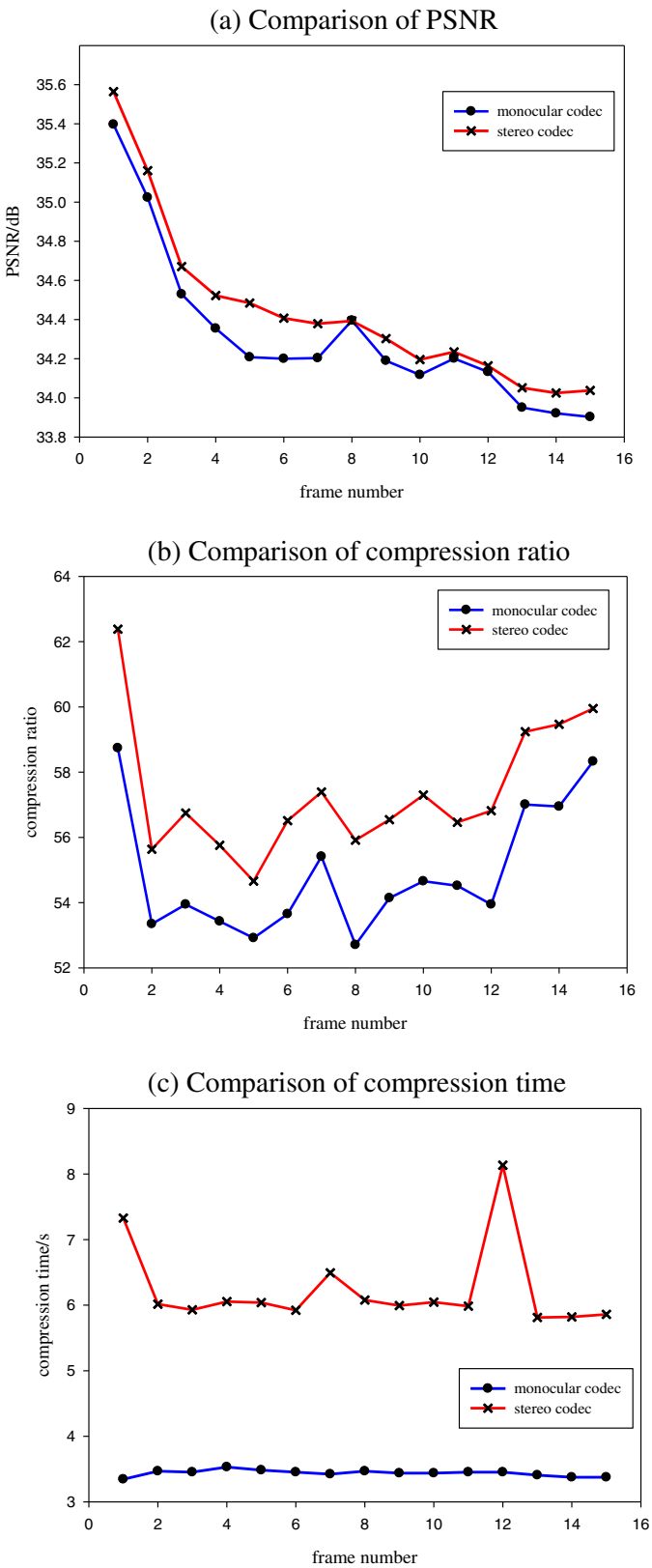


Figure 7 The comparison of proposed monocular codec and proposed stereo codec of “flamenco_l.yuv”. (a) Comparison of PSNR. (b) Comparison of compression ratio. (c) Comparison of compression time.

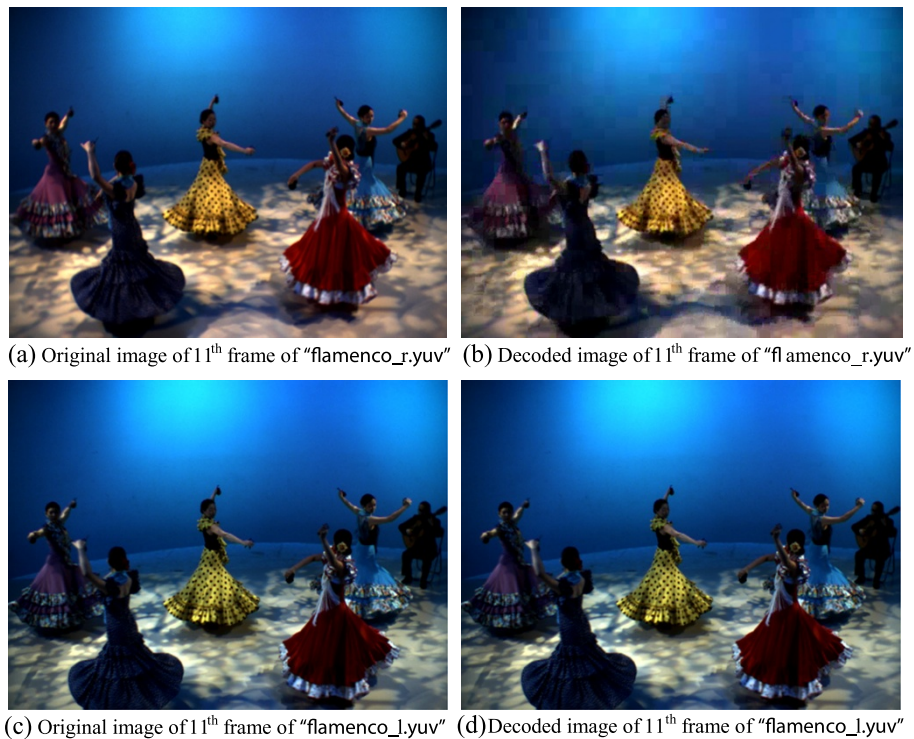


Figure 8 The decoded results of 11th frame of “flamenco_r.yuv” and “flamenco_l.yuv”. (a) Original image of 11th frame of “flamenco_r.yuv”. (b) Decoded image of 11th frame of “flamenco_r.yuv”. (c) Original image of 11th frame of “flamenco_l.yuv”. (d) Decoded image of 11th frame of “flamenco_l.yuv”.

Table 3 Performance comparison between proposed monocular video coding, proposed stereo video coding, JMVC (4.0)

Sequences 640 × 480 pixels	Methods	PSNR (dB) (average)	Bit rate/kbps	compression time (s/frame) (average)
Ballroom	Proposed monocular video coding	33.93	172.80	1.79
	Proposed stereo video coding	34.25	168.38	9.49
	JMVC (4.0)	33.45	187.76	12.07
Exit	Proposed monocular video coding	36.54	57.57	5.30
	Proposed stereo video coding	36.67	55.85	11.71
	JMVC (4.0)	36.03	88.10	10.36
Vassar	Proposed monocular video coding	35.09	52.37	3.53
	Proposed stereo video coding	35.12	50.93	8.49
	JMVC (4.0)	34.49	62.73	8.50

PSNR or the quality of decoded image in the proposed method could be improved by inserting the I-frame.

Table 2 shows the comparison of performances of the proposed method and the state-of-the-art fractal compression methods in [22,23]. The sequence “conference” (255 × 255 pixels, 15 frames) is used and the compression ratio is 6.13 times higher than that of [22] and 4.24 times higher than [23]. The proposed method achieves 98.8 % computational time saving with 0.9 dB higher of PSNR, comparing with the results of [22]. Although the PSNR of our scheme is decreased by 0.6 dB compared to that of [23], the coding time is saved by 98.7 %.

Stereo video coding

To evaluate the performance of the proposed stereo codec, we use “flamenco_r.yuv” and “flamenco_l.yuv” (640 × 480 pixels, 15 frames). The “flamenco_r.yuv” is right view, and the “flamenco_l.yuv” is left view. First,

Table 4 The performance comparison of NOB, OB, and H.264 of “foreman.cif”

	NOB	Object 1	Object 2	H.264
PSNR (dB)	34.56	35.23	36.65	35.96
Compression ratio	55.83	118.81	92.90	113.75
Compression time (s)	0.87	0.66	0.75	0.96

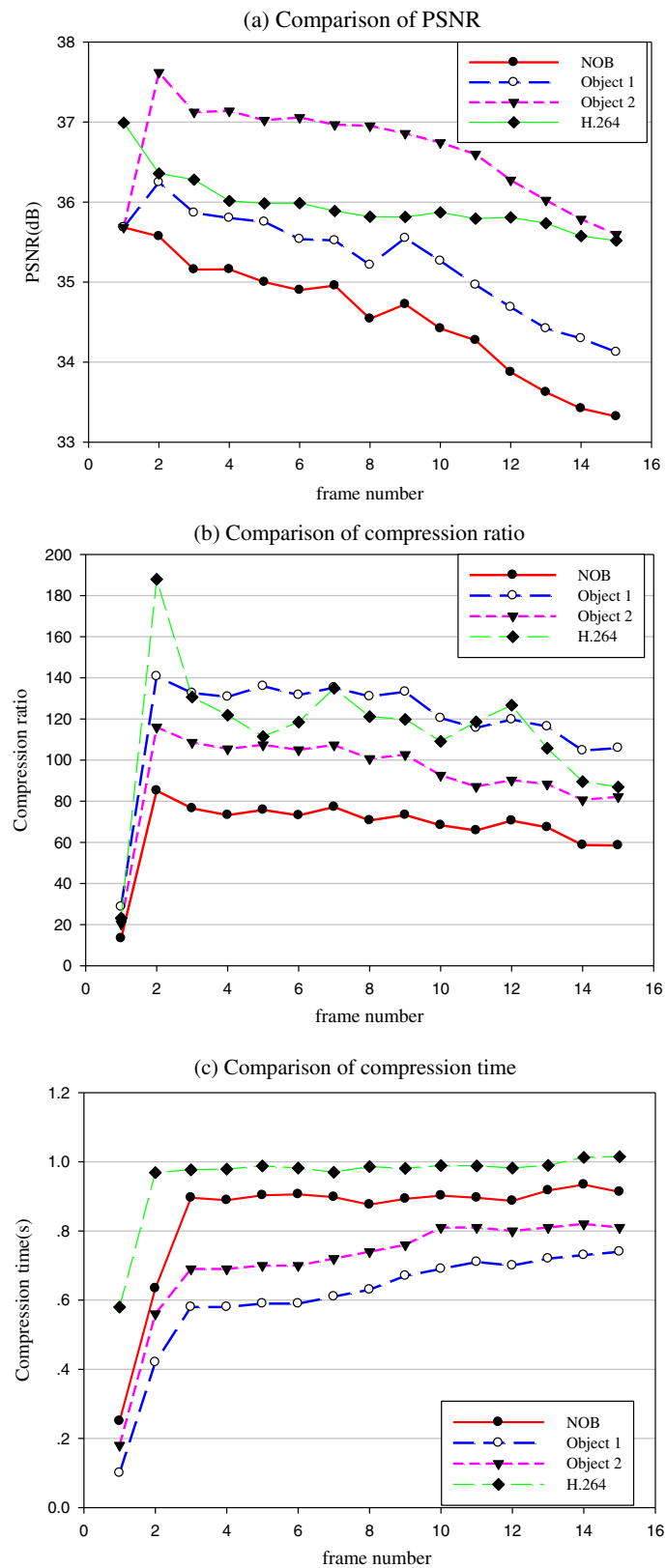


Figure 9 The comparison of the results of NOB and OB by the proposed method and by H.264 of "foreman.cif". (a) Comparison of PSNR. (b) Comparison of compression ratio. (c) Comparison of compression time

“flamenco_l.yuv” is compressed by monocular codec; second, it is compressed by stereo codec. As shown in Figure 7, the PSNR and compression ratio of stereo codec are better than monocular codec, as the calculation raises, the compression time is more than monocular codec. For example, the proposed stereo method can raise compression ratio 1.7 to 3.7 and improve the image quality 0.13 dB on average comparing to the proposed monocular method.

The decoded images of 11th frame of “flamenco_r.yuv” and “flamenco_l.yuv” are shown in Figure 8. Figure 8a is the original image of “flamenco_r.yuv”; Figure 8b is the decoded image of “flamenco_r.yuv” (compression ratio: 58.40, PSNR: 34.26 dB); Figure 8c is the original image of “flamenco_l.yuv”; Figure 8d is the decoded image of “flamenco_l.yuv” (compression ratio: 54.51, PSNR: 34.20 dB).

Table 3 shows the experimental results based on a set of stereo video sequences (“ballroom”, 640 × 480 pixels; “exit”, 640 × 480 pixels; “vassar”, 640 × 480 pixels) by comparing the proposed stereo video coding with the proposed monocular video coding and the JMVC full search (PelBlockSearch: PBS) [24]. The experiments are carried out by JMVC 4.0 (QP = 32) [25]. Two hundred

and forty-eight frames from each sequence are tested and the average values are listed in Table 3. As shown in Table 3, proposed stereo codec, compared to proposed monocular codec and JMVC 4.0, can achieve a certain enhancement in PSNR and reduction in bit rate. For example, the PSNR of the proposed stereo video coding is about 0.17 dB higher than that of the proposed monocular video coding, and 0.69 dB higher than that of JMVC 4.0 on average. Comparing with the bit rate resulted by the proposed monocular video coding and JMVC 4.0, the proposed stereo video coding achieves, on average, 2.53 and 21.14 Kbps bit rate saving, respectively. The compression time of stereo video coding is about 0.51 s less than JMVC 4.0 on average, but as the calculation raises, the compression time is more than monocular codec.

Object-based video coding

To evaluate the performance of the proposed OB codec, we use “foreman.cif” and its *alpha* plane. The study in [26] indicates that the encoding cost of *alpha* plane, which is about 0.021 bits per pixel, is much low. For the *alpha* plane of the sequence “foreman.cif”, its encoding cost is about 0.26 kb per image. Comparing to 148.5 kb



Figure 10 The decoded results of 9th frame of “foreman.cif”. (a) Original 9th frame of foreman.cif. (b) Decoded NOB 9th frame. (c) Decoded OB 9th frame foreground (Object 1). (d) Decoded OB 9th frame background (Object 2).

of each original image, we can ignore the additional bits for compression ratio. As shown in Table 4 and Figure 9, the average performance results of OB are better than NOB and H.264.

The decoded images of 9th frame of "foreman.cif" are shown in Figure 10. Figure 10a is the original image; Figure 10b is the decoded image by NOB (compression ratio: 65.77); Figure 10c is the decoded image of object 1 by OB (compression ratio: 115.66); Figure 10d is the decoded image of object 2 by OB (compression ratio: 87.13).

Conclusion

Based on the classical fractal video compression method, monocular and stereo fractal video compression methods are proposed in this article, experimental results indicate that the proposed monocular fractal video compression method can raise compression ratio 3.6 to 7.5 times, speed up compression time 5.3 to 22.3 times, and improve the image quality 3.81 to 9.24 dB in comparison with CPM/NCIM. The PSNR of the proposed stereo video coding is about 0.17 dB higher than that of the proposed monocular video coding, and 0.69 dB higher than that of JMVC 4.0 on average. Comparing with the bit rate resulted by the proposed monocular video coding and JMVC 4.0, the proposed stereo video coding achieves, on average, 2.53 and 21.14 Kbps bit rate saving, respectively. A new object-based method improves the performances of fractal video coding algorithm obviously. The proposed object-based fractal video coding method which can increase compression ratio, the decoded image quality, and speed is simple and effective, and adds more flexibility and practicability to the applications of fractal video coding.

Competing interests

The authors declare that they have no competing interests.

Acknowledgments

The project was sponsored by the National Natural Science Foundation of China (NSFC) under Grant nos. 61075011 and 60675018, and also the Scientific Research Foundation for the Returned Overseas Chinese Scholars from the State Education Ministry of China. The authors thank for their financial supports. The authors would like to thank the editors and the reviewers for their hard work and their insightful suggestions, which help improving this article.

Received: 3 April 2012 Accepted: 1 October 2012

Published: 25 October 2012

References

1. B B Mandelbrot, *The Fractal Geometry of Nature* (W H Freeman and Company, New York, 1982)
2. M Demers, H Kunze, Fractal-based methods in imaging. *Nonlinear Anal. Theory, Methods Appl* **71**(12), 1413–1419 (2009)
3. Y-L Lin, W Ming-Sheng, An edge property-based neighborhood region search strategy for fractal image compression. *Comput. Math. Appl.* **62**(1), 310–318 (2011)
4. M S Lazar, L T Bruton, Fractal block coding of digital video. *IEEE Trans. Circuits Syst. Video Technol.* **4**(3), 297–308 (1994)
5. M Wang, *Cuboid Method of Fractal Video Compression* (Macquarie Scientific Publishing, Sydney, 2004)
6. K U Barthel, T Voyer, Three-dimensional fractal video coding, in *IEEE International Conference on Image Processing*, vol. III, ed. by (Washington, DC, 1995), pp. 260–263
7. Y Fisher, T P Shen, D Rogovin, Fractal (self-VQ) encoding of video sequences, in *Proceedings of the SPIE Visual Communications and Image Processing*, vol. 2308, ed. by (Chicago, IL, 1994), pp. 1359–1370
8. C S Kim, S U Lee, Fractal coding of video sequence by circular prediction mapping, in *NATO ASI Conference on Fractal Image Encoding and Analysis*, vol. 5, ed. by (London, 1997), pp. 1–15
9. M Wang, C-H Lai, Grey video compression methods using fractals. *Int. J. Comput. Math.* **84**(11), 1567–1590 (2007)
10. M Gharavi-Alkhasani, T S Huang, Fractal video coding by matching pursuit, in *Proceedings of the IEEE International Conference on Image Processing*, vol. 1, ed. by (Lausanne, Switzerland, 1996), pp. 157–160
11. Y Wang, J Ostermann, Y Q Zhang, *Video Processing and Communication* (Prentice-Hall, 2002)
12. K Rob, *Overview of the MPEG-4 Standard* (2002). ISO/IEC JTC1/SC29/WG11 N4668
13. K Belloulata, J Konrad, Fractal image compression with region-based functionality. *IEEE Trans. Image Process.* **11**(4), 351–362 (2002)
14. B Peng, L Zhang, D Zhang, Automatic image segmentation by dynamic region merging. *IEEE Trans. Image Process.* **20**(12), 3592–3605 (2011)
15. A E Jaquin, Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Trans. Image Process.* **1**(1), 18–30 (1992)
16. A Jaquin, Fractal image coding: a review. *Proc IEEE* **10**, 1451–1465 (1993)
17. Y Fisher, Fractal encoding with quadrees. *Fractal Image Compression, in Theory and Applications to Digital Images*, ed. by Y. Fisher (Spring-Verlag, Berlin, 1995), pp. 55–77
18. G Oien, S Lepsoy, Fractal-based coding with fast decoder convergence. *Signal Process.* **40**(1), 105–117 (1994)
19. C S Kim, S U Lee, Fractal coding of video sequence using circular prediction mapping and noncontractive interframe mapping. *IEEE Trans. Image Process.* **7**(4), 601–605 (1998)
20. Y-M Zhou, C Zhang, Z-K Zhang, An efficient fractal image coding algorithm using unified feature and DCT. *Chaos. Solitons Fract.* **39**(4), 1823–1830 (2009)
21. M G Strintzis, S Malassiotis, Object-based coding of stereoscopic and 3D image sequences. *IEEE Signal Process. Mag.* **16**(3), 14–28 (1999)
22. M Q Wang, L Choi-Hong, A hybrid fractal video compression method. *Comput. Math. Appl.* **50**(3–4), 611–621 (2005)
23. M Q Wang, R Liu, L Choi-Hong, Adaptive partition and hybrid method in fractal video compression. *Comput. Math. Appl.* **51**(11), 1715–1726 (2006)
24. T X-L, D. S-k, C C-h, An analysis of TZSearch algorithm in JMVC, in *2010 International Conference on Green Circuits and Systems (ICGCS)*, ed. by (Shanghai, China, 2010)
25. S Heiko, H Tobias, S Karsten, *JMVC software model Version 4.0* (, 2009)
26. L Labelle, D Lauzon, J Konrad, E Dubois, Arithmetic coding of a lossless contour-based representation of label images, in *IEEE International Conference on Image Processing*, vol. 1 (Chicago, IL, USA, 1998), pp. 261–265

doi:10.1186/1687-6180-2012-227

Cite this article as: Zhu et al.: A novel fractal monocular and stereo video codec with object-based functionality. *EURASIP Journal on Advances in Signal Processing* 2012 **2012**:227.