

Hybrid 3D Fractal Coding with Neighbourhood Vector Quantisation

Zhen Yao

*Computer Science Department, University of Warwick, Coventry CV4 7AL, UK
 Email: yao@dcs.warwick.ac.uk*

Roland Wilson

*Computer Science Department, University of Warwick, Coventry CV4 7AL, UK
 Email: rgw@dcs.warwick.ac.uk*

Received 31 August 2003; Revised 12 September 2004

A hybrid 3D compression scheme which combines fractal coding with neighbourhood vector quantisation for video and volume data is reported. While fractal coding exploits the redundancy present in different scales, neighbourhood vector quantisation, as a generalisation of translational motion compensation, is a useful method for removing both intra- and interframe coherences. The hybrid coder outperforms most of the fractal coders published to date while the algorithm complexity is kept relatively low.

Keywords and phrases: fractal, compression, video coding, neighbourhood vector quantisation, convergence.

1. INTRODUCTION

Fractal image compression techniques, introduced by Barnsley and Jacquin [1, 2], are the product of the study of iterated function systems (IFS). These techniques involve an approach to compression quite different from standard transform coder-based methods. Transform coders model images in a simple fashion, as vectors drawn from a wide-sense stationary random process and store images as quantized transform coefficients. Fractal block coders, assume that image redundancy can be efficiently exploited through self-similarity on a blockwise basis. They represent images by contraction maps, of which the images are approximate fixed points. Images are decoded by iterating these maps to their fixed points.

The fundamental principle of fractal coding is to represent the image by a set of contractive mappings. First, the image I is partitioned into a set of blocks $\mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n\}$ that covers I , referred to as the *range blocks*. For each $\mathbf{r}_i \in \mathbf{R}$, a domain block $\mathbf{d}_j \in \mathbf{D}$, usually twice as large as the range block, is sought which most resembles \mathbf{r}_i after a contractive transform involving operations like rotation, and contrast and brightness adjustments. At its simplest, the affine mapping can be expressed as

$$\phi(\mathbf{x}) = s\mathbf{x} + \mathbf{o}, \quad |s| \leq 1, \quad (1)$$

where s is the scaling coefficient and \mathbf{o} is the offset coefficient. Hence a representation of the range block can be expressed

by the domain block index and s and \mathbf{o} coefficients. The best-matching domain block can be found by a systematic search, while the latter two can be directly optimised in a least squares sense.

The transform from the domain block to range block forms the contractive mapping ϕ_i for \mathbf{r}_i and the collection

$$\Phi = \{\phi_1, \phi_2, \dots, \phi_n\} \quad (2)$$

is the fractal coded representation of the image, also referred to as partitioned iterative function system (PIFS).

While fractal image compression has been studied in a large body of published literature, fractal-based techniques have also been explored for coding image sequences. They are usually divided into two categories: single-frame-based schemes and volume-based schemes. In single-frame-based schemes, compression is still done in a frame-by-frame basis as in the conventional fractal image coding, but employs methods similar to motion compensation to exploit the temporal redundancy. Volume-based schemes treat the image sequence as a 3D volume, and extend the 2D blocks into 3D “cubes.” In this section, we give a brief overview of the previous work done in this area.

1.1. Singled frame-based schemes

In 1992, Hurd et al. [3] published results on fractal-based video compression claiming compression ratios from 21 : 1 (average PSNR of 39.2 dB) to 79 : 1 (average PSNR

of 30.8 dB) for a 160×120 -grey-scale sequence. In their method, they encode the first frame using a regular fractal coder. For the following frames, they use the previous frame as the source of domain blocks. To approximate each range block in one frame they either (1) apply motion compensation and find a matching same-size domain block from previous frame or (2) find a single-matching larger size domain block with a contractive transformation applied on it from the previous frame. As the coding of this method is causal, the decoding process is noniterative, and yields a very fast decompression.

Fisher et al. [4] described a similar method to encode the image frames based on quadtree partition. He reported a compression ratio of 25 : 1 to 244 : 1, while the compression time is as low as 2.4–66 s/frame.

In 1993, Hürtgen and Büttgen [5] applied fractal techniques for low-bit-rate video coding. They used prediction by frame difference with no motion compensation. Then for each frame, they applied the fractal transform only to those regions where prediction failed. For range blocks located in those regions, the whole domain in the same frame was searched, in contrast to the previous approach. The 352×288 -Miss America video sequence was reported to be coded at 128 Kbps with PSNR of 36–37 dB, and at 64 Kbps with PSNR of 34–35 dB, and at 32 Kbps with PSNR of 30–32 dB. As the domain blocks for each range block were selected from the same frame, the decoder is iterative in this method.

This approach is more like with vector quantisation (VQ) rather than fractal, since the matching of blocks (vectors) are usually sought from a codebook derived from the previous frame. However, the scheme retains many of the features of a fractal method, including *spatial* resolution independence and computationally simple decoding. The performance of the scheme also demonstrated to be competitive.

1.2. Volume-based schemes

In 1994, Lazar and Bruton [6] extended Jacquin's 2D algorithm to 3D, and used 3D range and domain blocks for image compression. They also used a 3D block splitting method (it is based on quadtree partition, but slightly modified to partition the temporal dimension) and the search for selecting domain blocks is done within the neighbourhood of the range block. They reported an average compression ratio of 74 : 1 at a PSNR of 32 dB.

Chabarchine and Creutzburg reported a scheme [7] that also extends 2D fractal coding to 3D. The proposed method uses simple 2-level partition represented in an oct-tree at depth 1, while each cube is a possible domain block for its 8 subcubes as range. This resulted in very fast encoding and decoding, but with relatively poor reconstruction quality.

Since the volume-based approach is the direct extension of the 2D fractal image PIFS coding, it is both spatial and temporal resolution-independent. This means a compressed video can be decoded into arbitrary frame rates. Alternatively, the image frames can be subsampled in order to reduce

the encoding time and bit rate, and decoded into the original frame rate using fractal interpolation.

2. NEIGHBOURHOOD VECTOR QUANTISATION

Fractal coding is generally considered as a special VQ method. Instead of having an external vector codebook as side information, the codebook is self-contained as a set of contractive mappings. Such nature is sometimes called *self-quantising*.

The self-referencing mechanism of fractal coding suggests that only images with high redundancy can be efficiently coded. Consider images such as a chessboard pattern, which cannot be efficiently compressed by fractal coding but with an optimised VQ codebook, it still can be coded at a decent rate. However, most natural images are highly redundant and exhibit strong *local coherence*, which means a pixel is usually similar to its neighbourhood pixels. Especially in video, the temporal frame-to-frame coherence has motivated the development of motion estimation and compensation in the past decades.

The majority of motion estimation/compensation algorithms implicitly assume an image model based on the following relation:

$$I_n(\mathbf{x}) = I_m(F(\mathbf{x})), \quad (3)$$

where $I_n(\mathbf{x})$ represents the grey-level value at pixel position \mathbf{x} in the image n of a sequence. $m = n \pm 1$ depending on whether the direction of estimation is *backward* or *forward*.

The model states that the content of the current image is related in some way to the contents of an adjacent image in the sequence by means of function F , this function being the motion model employed by the estimation algorithm. This is intuitively a sensible assumption in that a scene consists of the same objects whose position varies slowly over time. The motion vectors are used to predict the next frame in a self-referencing manner and the residual between the signal and the prediction is expected to be sparse. Motion compensation, essentially DPCM on frames, therefore assumes certain stochastic relations. The proposed *neighbourhood VQ* is a generalisation of translational motion compensation.

2.1. Definitions

Suppose the set to be quantised is $S = \{s_1, s_2, \dots, s_n\}$, where each s_i is an ordered pair $s_i = (\mathbf{x}_i, g_i)$, $g_i \in \mathbb{R}'$, which usually represents the mean grey-scale intensity or color information on s_i and \mathbf{x}_i is the position vector on a particular support and $X = \bigcup \mathbf{x}_i$ forms a set. We define two metric functions, the *spatial* distance metric on \mathbf{x}_i , $\mathbf{d} : X \times X \rightarrow [0, \infty)$ defined on a Cartesian lattice, and the *distortion* metric, $e : \mathbb{R}' \times \mathbb{R}' \rightarrow [0, \infty)$ on g_i .

Definition 1. Given a distance η , the *neighbourhood set* $\mathcal{N}_{o,\eta}$ of a particular $o = (\mathbf{x}_o, g_o) \in S$ is defined as the set $\mathcal{N}_{o,\eta} = \{y \mid \text{for all } y = (\mathbf{x}_y, g_y) \in S : \mathbf{d}(\mathbf{x}_o, \mathbf{x}_y) \leq \eta\}$.

Definition 2. The set $\mathcal{N}_{o,1}$ is called o 's *connected neighbourhood* if the distance η in Definition 1 is 1.

For $o \in S$ and its neighbourhood set (codebook) $\mathcal{N}_{o,\eta}$, the quantised $\hat{o} = (\mathbf{x}, \hat{g}_o)$ is given by $\hat{g}_o = g_y$, where a $y = (\mathbf{x}_y, g_y) \in \mathcal{N}_{o,\eta}$ such that for all $z \in \mathcal{N}_{o,\eta}$, $e(g_y, g_o) \leq e(g_z, g_o)$.

2.2. Related work

Despite motion compensation that falls within the broad principle of neighbourhood VQ, a similar scheme appeared in literature and is called predictive VQ, or vector DPCM [8]. It is essentially a predictive method operating on a vector basis. Instead of quantising the input vector into one of its neighbours, it encodes the residual error from the prediction, a linear combination of its neighbourhood vectors, using a pretrained codebook.

The idea of combining fractal and VQ has been exploited by a number of researchers. Davoine et al. [9] proposed a fractal scheme with triangulation with a VQ codebook. A similar scheme was reported by Hamzaoui et al. in [10]. Gharavi-Alkhansari and Huang [11] use a combination of fractal coding and basis block projection to compression still images as a generalisation of fractal block coding and VQ. For each domain block, they generate three pools of range blocks.

- (1) *Higher-scale adaptive basis blocks.* The standard range block pool from a fractal coder, that is, spatially averaged copies of the domain blocks, augmented by rotated and reflected versions.
- (2) *Same-scale adaptive basis blocks.* Generated by selecting regions of the image which are the same size as the domain block. They are selected causally only from encoded parts. It may also be augmented by rotations and reflections.
- (3) *Fixed basis blocks.* A fixed pool of basis blocks that is known by the encoder and decoder as side information. No constraints such as orthogonality or completeness to apply on the basis.

This scheme certainly is a generalisation of fractal block coding and VQ. When only higher-scale blocks are used, it is the standard fractal block coding, and when only fixed basis blocks are used, it is equivalent to VQ. The same-scale basis is a particular case of neighbourhood VQ, though no constraints were set on the neighbourhood distance η , which in this case can be thought as infinity. As might be expected, the algorithm is expensive computationally.

Furthermore, Kim et al. proposed a scheme [12] called “fractal vector quantizer” which generates the VQ codebook from the coarsely approximated image. Levy and Wilson [13] proposed a symmetric VQ codebook design in 3D wavelet domain using similar affine transforms in fractal coding such as scaling and rotation. By using an orthonormal wavelet representation, the conventional fractal coders are replaced by a combination of VQ with symmetry operations, and they achieved coding rate 0.031 bpp at 35.89 dB.

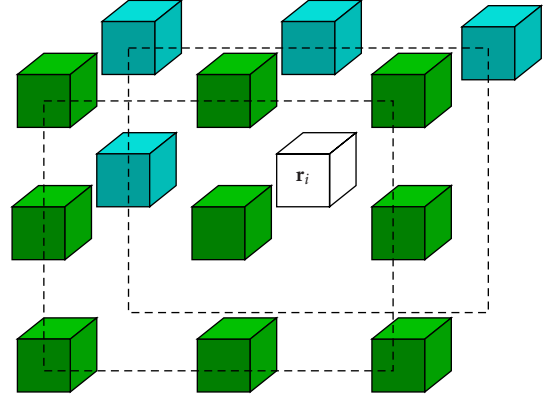


FIGURE 1: Connected neighbourhood for a 3D range block.

3. FRACTAL CODING WITH NEIGHBOURHOOD VQ

In this section, we present and discuss the design and implementation of the coder. We will also demonstrate a convergence problem in the proposed coder and how to overcome the defect.

3.1. Algorithm description

The baseline fractal coder is volume-based with the following configuration.

- (i) The support of the sequence volume is nonadaptively partitioned into $4 \times 4 \times 4$ range blocks.
- (ii) The domain search pool is selected *locally* near the range block, with domain block position increment of 2 pixels in order to reduce the total number blocks in the search pool.
- (iii) The number of transforms are extended to 16; the original 8 transforms proposed by Jacquin [2] and their time inverses.

The neighbourhood blocks (the virtual codebook) of a range block \mathbf{r}_i are causally selected from the connected neighbourhood; illustrated in Figure 1, 9 are in the previous time slice, and 4 are within the same time slice. Unlike the conventional motion compensation, where block matching is only done with a previous frame, neighbourhood VQ can exploit both temporal and spatial redundancy hence the local signal coherence can be well captured. For each $\mathbf{r}_i \in \mathbf{R}$, the range pool, we look for its best approximation \mathbf{r}_y in the *previously encoded, connected space neighbourhood* (see Definition 2) and their isometric transformed versions, where $e(\mathbf{r}_i, T_x(\mathbf{r}_y))$ is minimum. Essentially this forms a local symmetric codebook for the affine group T . If $e(\mathbf{r}_i, T_x(\mathbf{r}_y))$ is below a threshold σ , which suggests that \mathbf{r}_i and \mathbf{r}_y are similar, then \mathbf{r}_i is quantised as $T_x(\mathbf{r}_y)$. If $e(\mathbf{r}_i, T_x(\mathbf{r}_y)) > \sigma$, then \mathbf{r}_i is encoded using a conventional fractal contractive mapping.

In the implementation, instead of having the neighbourhood blocks collected from the original frames, we compare the range block with the blocks that were actually transmitted, that is, the previously quantised blocks, $\hat{\mathbf{r}}_y$. This prevents



FIGURE 2: Artifact generated by problematic convergence.

quantisation errors from accumulating and gives better rate-distortion performance. The error metric function $e(\cdot, \cdot)$ we used is the conventional squared-error measure.

3.2. Convergence improvement

It may not be obvious at first glance, that the previously proposed algorithm fundamentally changes the convergence condition of fractal coding. Fractal coding is said to be “eventually contractive” since the domain block under a contractive mapping is contractive itself, hence the rate of convergence for each mapping is actually faster than the contrast coefficient s suggests. This allows us to relax the convergence constraint $|s| < 1$ into $|s| \leq 1$ or even $|s| \leq 1.2$, which leads to improved reconstruction quality [14]. However, in the hybrid coder, due to strong local coherence, a large number of range blocks are *not* encoded with a contractive mapping but merely duplicated from some other range blocks. This can significantly slow down the convergence rate of the hybrid coder and sometimes it yields intolerable artifacts when the fractal-coded blocks are too sparse, typically when a range block is mapped from a domain block covered by its “offspring.” This is seen as a spread of defective blocks of identical luminance, as illustrated in Figure 2.

In order to eliminate the artifact, we need to increase the local convergence rate. The obvious way to do that is to set a tighter upper bound on $|s|$. However, it would degrade the reconstruction quality too much and not all the fractal-coded blocks need to lie within the constraint. The principle of our solution is to detect the potential blocks in which such defect can occur and then force one of their offspring to be fractal-coded, in order to reduce the local sparsity of fractal-coded blocks.

For the sake of implementation simplicity, though the domain block \mathbf{d} can overlap with $3 \times 3 \times 3 = 27$ blocks, we only consider 8 of them in a $2 \times 2 \times 2$ -square region. We denote these 8 range blocks as $\{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \dots, \mathbf{r}_8\}$, spatially arranged as in Figure 3. Then $(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4)$ forms the *horizontal* plane, $(\mathbf{r}_1, \mathbf{r}_3, \mathbf{r}_5, \mathbf{r}_7)$ forms the *vertical* plane, and

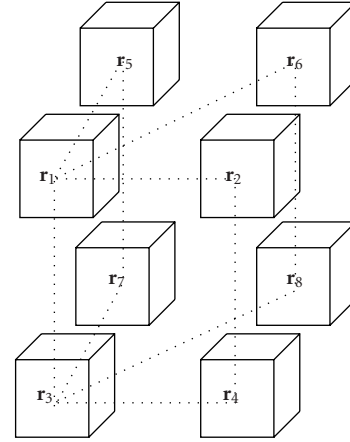


FIGURE 3: Spatial arrangement for range blocks.

$(\mathbf{r}_1, \mathbf{r}_3, \mathbf{r}_6, \mathbf{r}_8)$ forms the *diagonal* plane. Clearly the union of these planes covers the whole square. On each plane, we check to see if the range blocks on that plane are duplicated from the same block or the range block itself. The plane is *uniform* if it is true. We force \mathbf{r}_2 in horizontal plane, \mathbf{r}_7 from vertical plane, and \mathbf{r}_8 from the diagonal plane to be fractal-coded if the corresponding plane is uniform in order to increase the local density of fractal-coded range blocks.

3.3. Rate control

Rate control can be provided in various ways. The granularity of the quantisation on the scaling factor s and offset factor o as well as the radius of the domain search range are obvious parameters for variation, and the rate-distortion effects of changing these parameters have been widely studied in the literature. However, using different search range and block partition sizes is not recommended for controlling the bit rate. The reason for not using search range as a control factor is coding speed. We reject the possibility of using different sizes of partition blocks because the reconstruction quality of using larger partition sizes such as $8 \times 8 \times 8$ is not acceptable. In the hybrid coder, rate control is primarily achieved by choosing different threshold σ values, as will be shown in the following section.

4. EXPERIMENTAL RESULTS

Based on a standard setting of $4 \times 4 \times 4$ -block partition, a search range of 2, 4 bits for s and 5 bits for o , we tested the hybrid coder on video sequence *missa* (see Table 1) and medical volume *chest* (see Table 2) with four different configurations:

- (i) 16 transforms with no fractal interpolation,
- (ii) 8 transforms with no fractal interpolation,
- (iii) 16 transforms with fractal interpolation on subsampled sequence of odd-numbered frames,
- (iv) 8 transforms with fractal interpolation on subsampled sequence of odd-numbered frames.

TABLE 1: Results from the hybrid volume coder on *missa*.

σ (dB)	$ T $	Interpolation	Rate (bpp)	PSNR (dB)
24	16	No	0.090	31.451
38	16	No	0.142	35.490
24	8	No	0.074	31.210
38	8	No	0.128	35.704
24	16	Yes	0.042	31.755
38	16	Yes	0.068	34.449
24	8	Yes	0.040	30.960
38	8	Yes	0.070	34.444

TABLE 2: Results from the hybrid volume coder on *chest*.

σ (dB)	$ T $	Interpolation	Rate (bpp)	PSNR (dB)
24	16	No	0.090	32.690
38	16	No	0.136	36.363
24	8	No	0.075	32.662
38	8	No	0.125	36.289
24	16	Yes	0.050	31.646
38	16	Yes	0.075	35.117
24	8	Yes	0.043	31.684
38	8	Yes	0.090	34.922

In the configurations with 16 isometric transforms, as mentioned before, the extra 8 transforms are time-reverses of the basic transforms as in [2]. These time-reverses are not present in the configurations with 8 transforms. The obvious reason for having less transforms is to gain coding speed: a coder with 8 transforms is at least two times faster than one with 16, since the search spaces for both neighbourhood VQ and fractal coding are reduced. We also expect around 1 bit per range block rate saving from the configuration of 8 transforms (since $\log_2 16 - \log_2 8 = 1$), as can be seen on the rate-distortion curves (Figure 4) by comparing the distance on rate axis between the marked points with same σ values. Interestingly, the impact of having fewer transforms on rate-distortion is not as severe as we expected. At lower bit rates, the version with 8 transforms constantly outperforms the one with 16 transforms. As the bit rate increases, the later version will eventually achieve superiority. However, even when the version with 16 transforms outperforms the 8-transform, their rate-distortion is very close and the 8-transform clearly is preferable due to its faster encoding time.

We also observe that in volume data *chest* the intersections of the two curves come earlier than in video data *missa*. This is due to the fact that temporal redundancy in video sequences is very *orientational*, resulting in time-inversed transforms being seldom used. However in a volume data they are more useful in finding the best approximation since the redundancy is less orientational.

Fractal interpolation on the temporal direction was also examined. We subsampled the sequences frames by a factor of 2, and operate the hybrid coder on the sampled frames, then decompress into the original frame number. It

was shown that very impressive compression performance was achieved by this approach, typically at 0.05 bpp with 34 dB on video and at 0.07 bpp with 35 dB on volume data comparing with the original sequences. Such interpolation property is desired particularly with medical volume data, when details need to be enlarged with certain faithfulness in order to reveal some subtle details from the compressed representation. It could also accelerate the volume rendering process by decompressing the sequence into a smaller size.

Since performing neighbourhood VQ is much faster than fractal coding, The encoding speed of the hybrid coder is also promising, typically 3–6 frames/s on a 600 MHz Pentium II processor, comparable to a standard MPEG-2 encoder. However, it should be noted that it is hard to benchmark the speed performance since the encoding time varies quite significantly with different coder settings and also very data-dependent. The computation time is approximately linear with the number of fractal-coded blocks. Setting a high σ threshold would obviously speed up the coding process, but with a trade-off on reconstruction quality. With fractal interpolation, we can essentially reduce the amount of data, which can increase the speed significantly. Generally speaking, although the performance may not outperform DCT-based algorithms, it is not far worse than them and significantly faster than its fractal counterparts. With a local search pool, the coding delay is typically 16 frames, approximately half a second for video sequences. Although the delay is longer than MPEG video coding standards, which usually only requires 1 or 2 frames to perform motion estimation, the delay will not cause severe propagation in video transmission.

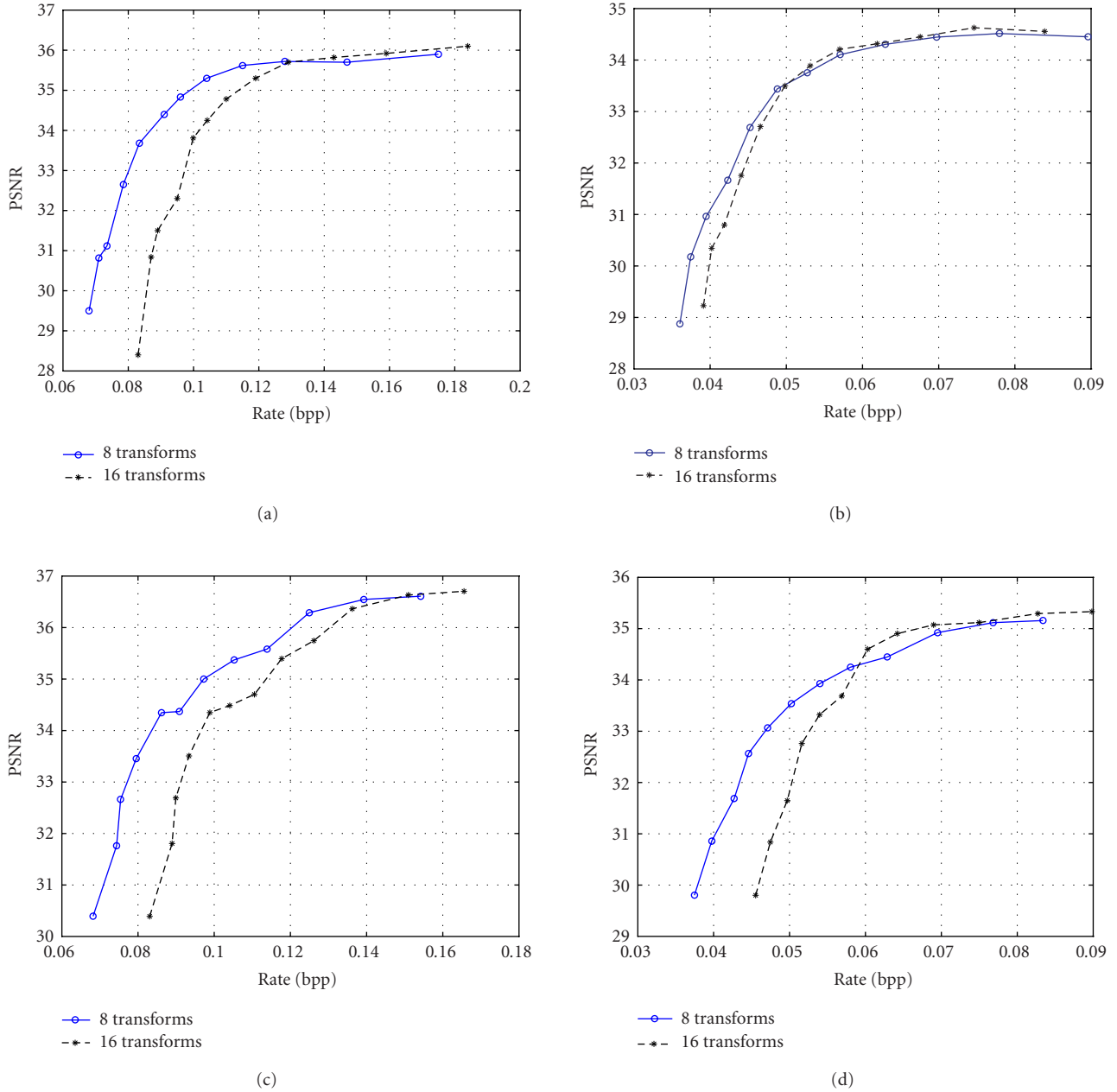


FIGURE 4: (a) Rate-distortion curve of the hybrid volume coder without fractal interpolation on *missa* sequence. (b) Rate-distortion curve of the hybrid volume coder with fractal interpolation on *missa* sequence. (c) Rate-distortion curve of the hybrid volume coder without fractal interpolation on *chest* sequence. (d) Rate-distortion curve of the hybrid volume coder with fractal interpolation on *chest* sequence.

Comparing with results from other proposed fractal-based coders on the *missa* sequence, Fisher's single-frame based scheme [4] encodes 0.126 bpp with 33.79 dB and 0.2 bpp with 35.74 dB. Our scheme yields 0.127 bpp at 35.4 dB and 0.185 bpp at 36.0 dB without interpolation, and 0.054 bpp at 33.80 dB with interpolation. Work [6] by Lazar and Bruton is volume based, achieving 0.107 bpp with 32 dB. It is beaten by our results of 0.090 bpp at 33.76 dB without interpolation and 0.045 bpp at 32 dB with interpolation.

MPEG-2 encodes the *missa* at 0.11 bpp with 34.93 dB, while 0.11 bpp from the hybrid fractal volume coder can offer 35.3 dB on PSNR without interpolation. Finally, though Levy-Wilson's result (0.031 bpp at 35.89 dB) outperforms significantly: most of the coding efficiency was due to the wavelet decomposition in their scheme, instead of using purely fractal PIFS. On the artifact assessment, the main artifacts of the reconstructed sequence are the block artifacts, as can be seen in Figures 5 and 6. At a very low bit



(a)



(b)

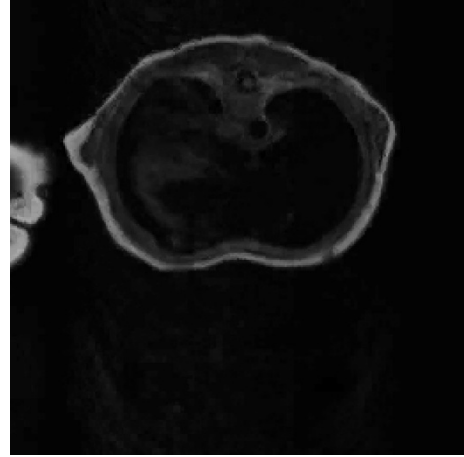
FIGURE 5: Example of reconstructions from frame 23 from *missa* with different coder settings. (a) $\sigma = 40$, $|T| = 16$, rate = 0.159 bpp, PSNR = 35.902 dB. (b) $\sigma = 32$, $|T| = 8$, interpolated, rate = 0.053 bpp, PSNR = 33.440 dB.

rate, quantisation effects on luminance are also visible, and “blinking pixels” can be seen in some occasions.

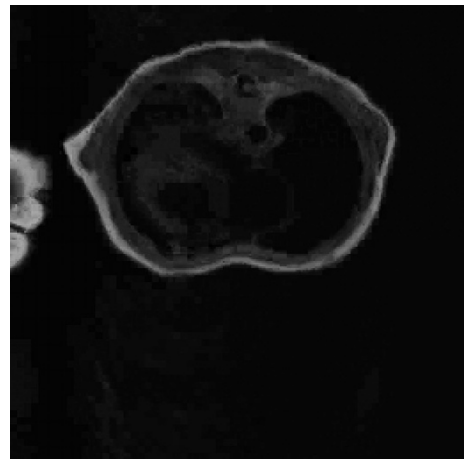
Despite these artifacts, since the coder operates on a block basis, the block artifact can be observed in the temporal dimension as well, when quality “jumps” between two blocks of frames. This is illustrated in Figure 7 where sharp slopes can be seen on the PSNR curve between individual frames.

5. VISUALISATION

It was suggested in [2] that fractal coding should be applied for sharp-edged blocks, whereas VQ would be more advantageous for other blocks such as in plain or textured areas.



(a)



(b)

FIGURE 6: Example of reconstructions from frame 55 from *chest* with different coder settings. (a) $\sigma = 40$, $|T| = 16$, rate = 0.151 bpp, PSNR = 36.632 dB. (b) $\sigma = 32$, $|T| = 8$, interpolated, rate = 0.054 bpp, PSNR = 33.929 dB.

In the 3D case, the fractal-coded block should then approximately follow the surfaces of volume objects or the swiping plain formed by 2D edge contours in video. In a compression paradigm, since a fractal-coded block requires more bit budget and time, we should be able to enhance the coding both computationally and in rate-distortion sense by reducing the number of fractal codes. We designed a visualisation tool in order to see how the hybrid coder selects the fractal-coded areas. It simply plots a cube in the locations which are being fractal-coded. Results demonstrated in Figure 8 are quite reassuring. The visualised blocks represent the structure of the sequence quite well. In Figure 8a we can see the figure of the person is well outlined, and in Figure 8b important surfaces of the *chest* are formed by those blocks.

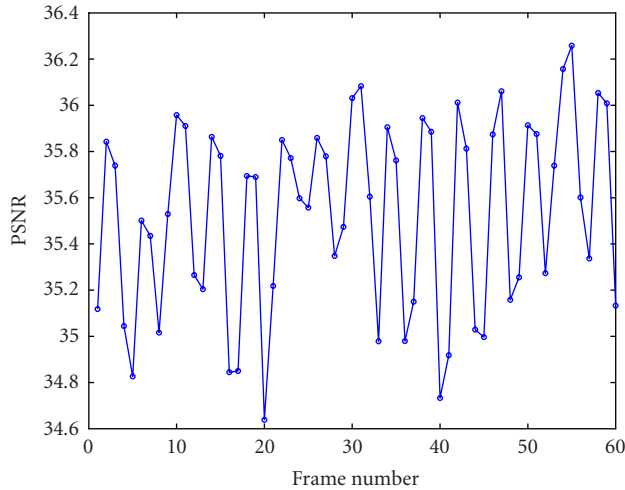
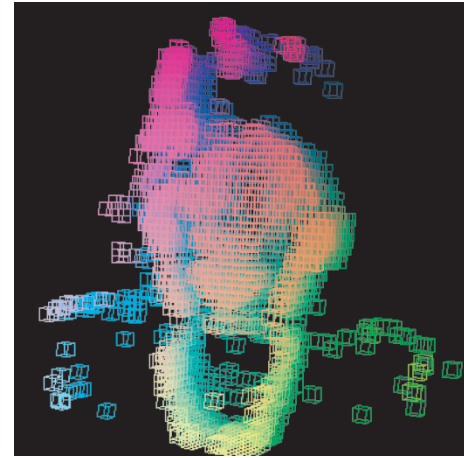


FIGURE 7: Individual frame PSNR = 35.32 dB for *missa* sequence, with $\sigma = 34$ dB at 0.118 bpp.

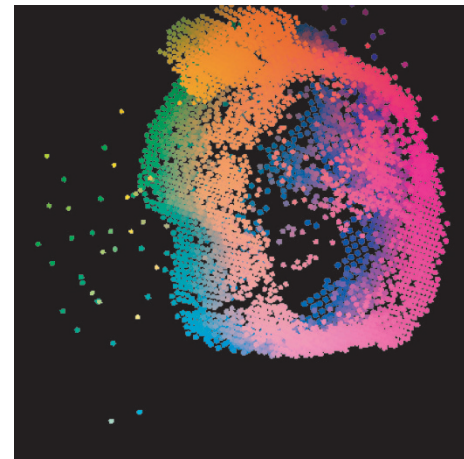
6. CONCLUSION

We have presented and discussed the concept of neighbourhood VQ and how it can be applied with conventional fractal coding to compress video. Its performance beats most of the fractal-based video coders published so far and is comparable with MPEG-2 standard with simpler complexity. However, this approach could be problematic because of the slower convergence rate and it is no longer *locally eventually contractive*. The effort of increasing local fractal-coded range blocks was empirically demonstrated to be successful, since the problematic convergence artifacts was never observed in the modified coders. Though the artifact is eliminated, this fix does not fundamentally increase the convergence rate. Slower convergence also implies more computation on iteration for the decoder to carry out.

While neighbourhood VQ fits well with the constant blockwise partition, the possibility of employing adaptive partition schemes such as quadtree partitions was not studied in this work. Certainly, the hybrid coders should gain significant benefits by using the quadtree (or oct-tree) partition [14]. However an immediate problem is the difficulty in establishing the neighbourhood codebook, since the surrounding blocks can be arbitrarily partitioned into very small blocks and the data structure representing the quadtree is recursive. Finding spatially close range blocks is much more difficult than it may seem. A possible solution is to adopt space-filling curves such as the Hilbert curve to traverse the partition. The Hilbert curve has the property that it will not leave a quadrant until each block in that quadrant has been visited exactly once. This will decompose the whole partition in a sequential manner. Past work shows that compressing the sequence decomposed by the Hilbert curve will asymptotically reach the entropy of the image using the LZ78 [15] coder. This shows such decomposition can be effective, since blocks that are spatially near in 2D support will be close in the sequence.



(a)



(b)

FIGURE 8: Visualisation results on (a) *missa* frame number 1–48 and (b) *chest* frame number 1–96.

REFERENCES

- [1] M. F. Barnsley, *Fractal Everywhere*, Academic Press, San Diego, Calif, USA, 2nd edition, 1993.
- [2] A. E. Jacquin, "Fractal image coding: a review," *Proceedings of the IEEE*, vol. 81, no. 10, pp. 1451–1465, 1993.
- [3] L. P. Hurd, M. A. Gustavus, and M. F. Barnsley, "Fractal video compression," in *Digest of Papers 37th IEEE Computer Society International Conference (COMPCON '92)*, pp. 41–42, San Francisco, Calif, USA, February 1992.
- [4] Y. Fisher, D. N. Rogovin, and T.-P. J. Shen, "Fractal (self-VQ) encoding of video sequences," in *Visual Communications and Image Processing*, A. K. Katsaggelos, Ed., vol. 2308 of *Proceedings of SPIE*, pp. 1359–1370, Chicago, Ill, USA, September 1994.
- [5] B. Hürtgen and P. Böttgen, "Fractal approach to low-rate video coding," in *Visual Communications and Image Processing*, B. G. Haskell and H.-M. Hang, Eds., vol. 2094 of *Proceedings of SPIE*, pp. 120–131, Cambridge, Mass, USA, November 1993.
- [6] M. S. Lazar and L. T. Bruton, "Fractal block coding of digital video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, no. 3, pp. 297–308, 1994.

- [7] A. Chabarchine and R. Creutzburg, "3D fractal compression for real-time video," in *Proc. 2nd IEEE International Symposium on Image and Signal Processing and Analysis (ISPA '01)*, pp. 570–573, Pula, Croatia, June 2001.
- [8] C. W. Rutledge, "Vector DPCM: vector predictive coding of color images," in *Proc. IEEE Global Telecommunications Conference*, pp. 1158–1164, Houston, Tex, USA, September 1986.
- [9] F. Davoine, M. Antonini, J.-M. Chassery, and M. Barlaud, "Fractal image compression based on delaunay triangulation and vector quantization," *IEEE Transactions on Image Processing*, vol. 5, no. 2, pp. 338–346, 1996.
- [10] R. Hamzaoui, M. Müller, and D. Saupe, "Enhancing fractal image compression with vector quantization," in *Proc. IEEE Digital Signal Processing Workshop*, pp. 231–234, Loen, Norway, September 1996.
- [11] M. Gharavi-Alkhansari and T. S. Huang, "Generalized image coding using fractal-based methods," in *Proc. International Picture Coding Symposium (PCS'94)*, pp. 440–443, Sacramento, Calif, USA, September 1994.
- [12] C.-S. Kim, R.-C. Kim, and S.-U. Lee, "A fractal vector quantizer for image coding," *IEEE Transactions on Image Processing*, vol. 7, no. 11, pp. 1598–1602, 1998.
- [13] I. K. Levy and R. Wilson, "Three-dimensional wavelet transform video coding using symmetric codebook vector quantization," *IEEE Transactions on Image Processing*, vol. 10, no. 3, pp. 470–475, 2001.
- [14] Y. Fisher, Ed., *Fractal Image Compression: Theory and Application*, Springer-Verlag, New York, NY, USA, 1995.
- [15] A. Lempel and J. Ziv, "Compression of two-dimensional images," in *NATO ASI Ser. F*, Z. Galil and A. Apostolico, Eds., vol. 12 of *Combinatorial Algorithms on Words*, pp. 141–154, June 1985.

Zhen Yao was born in Hangzhou, China, on August 2, 1981. He received the B.S. degree in computer science from the University of Warwick, United Kingdom, with First-Class Honors in 2003. He is currently pursuing a Ph.D. degree in computer science in the same institute. He was the recipient of the Best Student Paper Award from the IEEE Region 8 UKRI student paper contest in 2003.



Roland Wilson received the B.S. and Ph.D. degrees from the Department of Electrical and Electronic Engineering at the University of Glasgow, in 1971 and 1978, respectively. From 1978 to 1985, he was a Lecturer in the Department of Electronic and Electrical Engineering at the University of Aston. In 1982–1983, he was a Visiting Professor at Linköping University, Sweden. In 1985, he was appointed to a Senior Lectureship in the Department of Computer Science at the University of Warwick. In 1992, he was promoted to a Readership. In 1985, he was jointly awarded the Pattern Recognition Society Medal for Best Paper in *Pattern Recognition* with his student Mike Spann. In 1999, he was promoted to a Professorship. He has published over 100 papers in the areas of communication theory, image and audio signal processing, and neural networks. He is an Editorial Board Member for the journal *Pattern Recognition*.

