

# Multiple same-sized block mapping for recursive fractal image coding

**Yao Zhao**

Northern Jiaotong University  
Institute of Information Science  
Beijing 100044, China  
E-mail: y.zhao@its.tudelft.nl

**Hongxing Wang**

Beijing University of Aeronautics and  
Astronautics  
Beijing 100083, China

**Baozong Yuan**

Northern Jiaotong University  
Institute of Information Science  
Beijing 100044, China

**Abstract.** In conventional fractal image coding (FIC) schemes, domain blocks are constrained to be twice as large as range blocks to ensure the convergence of their iterative decoding stage. However, this constraint has limited the fractal encoder to exploit the self-similarity at the same resolution scale of natural images. To overcome the shortcoming, a novel scheme using same-sized range and domain blocks is proposed. Further, a recursive scheme feeding the coding results back to the input during the encoding procedure is used to improve the decoded image quality. Experimental results show our method gives significant improvement over Fisher's FIC. © 2002 Society of Photo-Optical Instrumentation Engineers. [DOI: 10.1117/1.1428741]

Subject terms: fractal image coding; image compression.

Paper 200447 received Nov. 13, 2000; revised manuscript received July 9, 2001; accepted for publication July 27, 2001.

## 1 Introduction

The concept of fractal image coding was proposed by Barnsley in 1987,<sup>1</sup> and the first automatic FIC scheme that compressed arbitrary monochrome images was proposed by Jacquin in 1990.<sup>2</sup> It modeled a natural image as a fractal picture and made use of the block-wise self-similarity of the image to obtain the parameters of fractal transform. The scheme was a milestone paper and it was followed by many publications varying and improving on Jacquin's basic idea.<sup>3-15</sup> Some papers improve the image quality by adopting different partitions.<sup>3-6</sup> Some combined FIC with transform coding methods.<sup>7-9</sup> And others tried to use nonlinear approximation in gray levels.<sup>10,11</sup> Fisher modified Jacquin's partition strategy and proposed a practical scheme that achieves better bitrate/image quality performance.<sup>3</sup>

In Jacquin's scheme,<sup>2</sup> the original image is first partitioned into two types of blocks called range and domain. The range blocks tile the whole image, and the arbitrarily located domain blocks are twice as large as range blocks. For every range block, a suitable domain block and affine transform are sought so that the transformed domain block is similar to the range block. The parameters of the affine transformation are quantized, thus the range blocks are encoded. The decoding procedure is comparatively simple. All the affine transforms are decoded and are iteratively applied on an arbitrary initial image. The fixed point and collage theorem guarantee the convergence.<sup>12</sup> We consider two problems to improve conventional fractal coders, namely: how to exploit similarity at the same resolution scale, and an extension to a collage theorem.

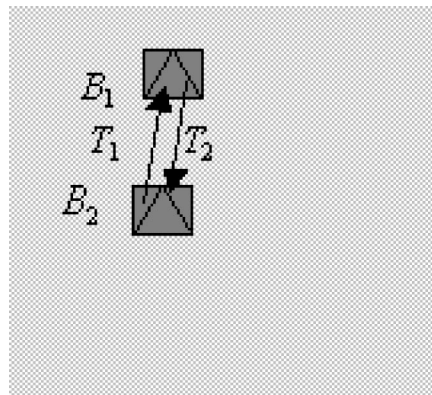
### 1.1 How to Exploit Similarity at the Same Resolution Scale

In all conventional FIC schemes, domain blocks are always constrained to be larger (usually twice the size) than range blocks to ensure the convergence of the iterative decoding procedure. Therefore, range blocks do not find similar

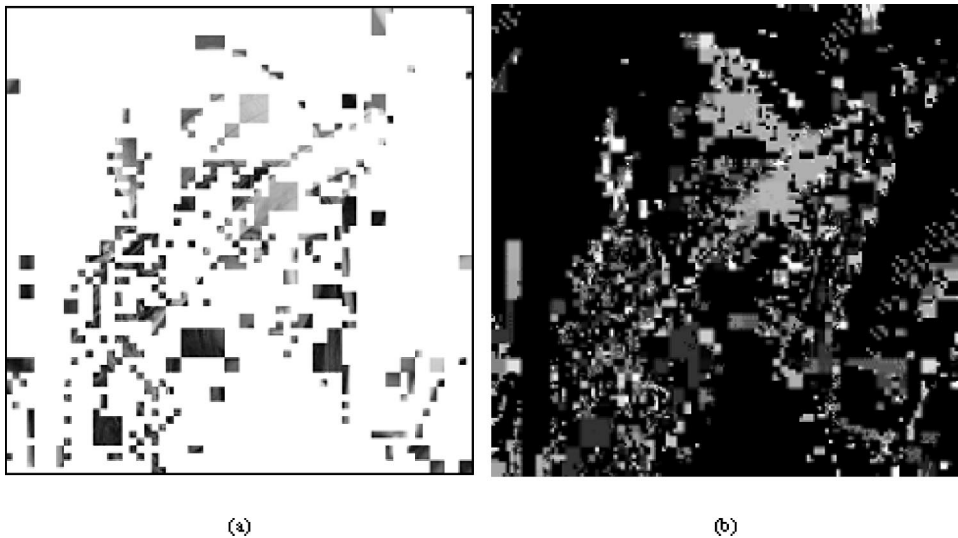
blocks at the same resolution scale in the original image. Instead, they search for the similar blocks in the down-sampled image. This means that the conventional FIC scheme only exploits the self-similarity of different scales.

However, in real natural images, there exist not only the similarity of a different scale but also similarity at the same scale. For example, the left eye of a person is very similar to the right, and they are of the same size instead of different sizes. Conventional FIC schemes cannot exploit such kind of self-similarity.

To overcome the drawbacks, using domain blocks of the same size as range blocks is an intuitive way. Bedford et al. first proposed this idea.<sup>15</sup> However, if we simply adopt their type of domains without condition, the decoding procedure cannot converge properly. Figure 1 illustrates such an example. In the image, same-sized blocks  $B_1$  and  $B_2$  are similar to each other. In the encoding procedure, the two blocks are mapped from each other. In such a case, the decoding cannot converge since the pixels' gray levels in the area of



**Fig. 1** A special case when adopting same-sized domain blocks in FIC.



**Fig. 2** (a) The white areas are encoded with same-sized mapping, others are encoded with twice larger domain blocks; and (b) the reconstructed image.

$B_1$  and  $B_2$  are unchanged during the decoding iterations. Bedford et al.<sup>15</sup> also mentioned the problem, but claimed they never met such cases. Unfortunately, such cases do occur frequently. Figure 2 illustrates a practical example using same-sized and twice larger domain blocks. Figure 2(a) shows which parts of the standard test image “Lena” are encoded with same-sized domain blocks and which parts are encoded with twice larger domains. Figure 2(b) shows the reconstructed image. Clearly the reconstructed image quality is very poor.

From this illustration, we can conclude that directly using same-sized domain blocks will cause the convergence problem. In Sec. 2, we propose a practical scheme adapting same-sized domain blocks and twice larger domain blocks.

## 1.2 Extension to Collage Theorem

The mathematical foundation of FIC is based on the fixed point theorem and collage theorem.<sup>12</sup> The collage theorem means that if the difference error between the original image and its collage is small enough, then the difference error between the decoded image and the original is also small. That is to say, the condition is a sufficient condition. However, it is not a necessary condition, since the difference between the original and the decoded image is not directly influenced by the difference between the image and its collage. So in the encoding procedure, minimizing the difference between the original image and its collage usually does not result in the minimizing of the difference between the decoded image and the original image. Figure 3 illustrates the collage images and decoded images of two FICs. Comparing the two FICs, we know that the quality of the collage image of FIC A is better than that of FIC B, however, the decoded image of FIC A is worse than FIC B.

Several researchers have noticed this problem and have proposed alternative solutions.<sup>12–14</sup> In these papers, the common idea is using an iterative coding scheme, i.e., the original image is encoded and decoded for several times. In each iteration, fractal transforms are achieved using con-

ventional FIC, and then the domain pool is updated for the next iteration with the decoded image. The procedure is shown in Fig. 4.

From Fig. 4, in every iteration, conventional fractal image coding and decoding are needed, and the iteration number is unknown in advance. Therefore, the main drawback of these schemes is their heavy computation burden.

To improve the performance without significantly increasing the computation burden, a novel coding method is proposed.

Section 2 describes the principle of multiple same-sized block mapping to exploit the similarity at the same scale. Section 3 proposes a recursive scheme to improve the image quality. Section 4 presents the implementation diagram combining these two ideas. Section 5 presents some experimental results, and conclusions are given in Sec. 6.

## 2 Multiple Same-Sized Block Mapping and Its Convergence

As discussed in Sec. 1, similarity at the same resolution most commonly exists in natural images, so it is desirable to use same-sized domain blocks in FIC. However, using such types of domain blocks without constraints will cause convergence problems. Our solution is illustrated in Fig. 5. The basic idea is to construct compound transforms to be eventually contractive, i.e., if a range block  $B_1$  is mapped with a same-sized block  $B_2$ , then another range block containing pixels of  $B_2$  must be mapped with a twice larger domain block  $B_3$ . In this case, the compound transform involved is contractive. The contractivity of such a case can be proved theoretically.

In FIC, every transform for a range block can be written as an affine transform.<sup>2–3</sup> In common sense, when two blocks are similar to each other, they are usually in the same direction. So in the same-sized mapping proposed here, the transform involved is only a position translation without rotation or flipping. In Fig. 5, if a range block is encoded with a same-sized domain block, the relation be-

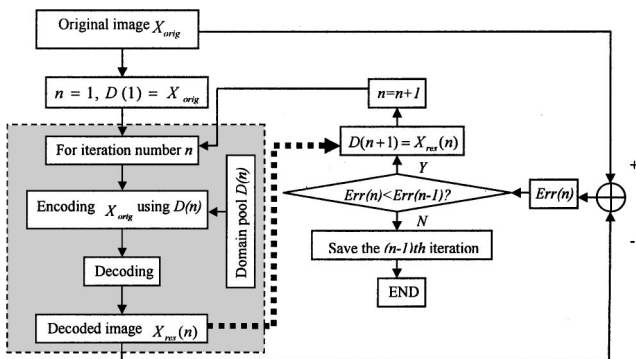


**Fig. 3** (a) The collage image of FIC A, PSNR=33.38 dB; (b) the decoded image of FIC A, PSNR=28.92 dB; (c) the collage image of FIC B, PSNR=32.15 dB; and (d) the decoded image of FIC B, PSNR=31.51 dB.

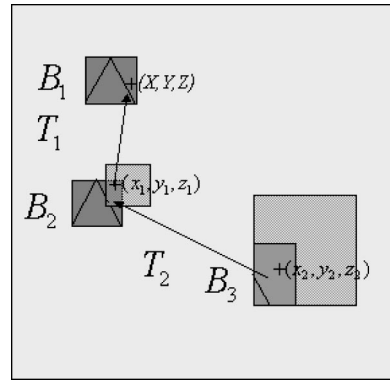
tween a point  $(X,Y,Z)$  in block  $B_1$  and another correspondent point  $(x_1,y_1,z_1)$  in block  $B_2$  can be expressed as an affine transform:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \alpha_1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \begin{bmatrix} e_1 \\ f_1 \\ 0 \end{bmatrix}, \quad (1)$$

where,  $(x_1,y_1)$  is a pixel position,  $z_1$  is the pixel gray value,  $(X,Y)$  is the transformed pixel position,  $Z$  is the transformed pixel value,  $e_1, f_1$  are the parameters of the transform denoting the position shift, and  $\alpha_1$  is the scale factor of gray level,  $0 \leq \alpha_1 \leq 1$ . The affine transform only makes a block shift in position without shrinking in size.



**Fig. 4** The core principle of some iteration coding schemes.



**Fig. 5** A compound transform that is eventually contractive.

In the case shown in Fig. 5, we know that the pixel  $(x_1,y_1,z_1)$  is mapped from another point by a contractive transform:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} a_2 & b_2 & 0 \\ c_2 & d_2 & 0 \\ 0 & 0 & \alpha_2 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ f_1 \\ o_2 \end{bmatrix}, \quad (2)$$

where the parameters  $a_2, b_2, c_2, d_2, e_2, f_2$  make the affine transform contractive in the X-Y plane,  $\alpha_2$  ( $|\alpha_2| < 1$ ) makes it contractive in the gray level, and  $o_2$  is a gray level offset.

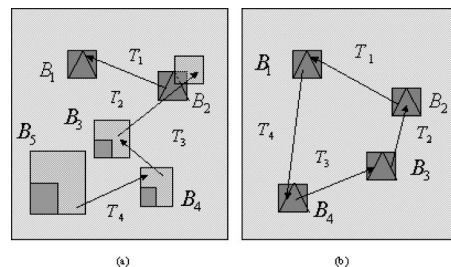
From Eqs. (1) and (2), we get:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \alpha_1 \end{bmatrix} \left( \begin{bmatrix} a_2 & b_2 & 0 \\ c_2 & d_2 & 0 \\ 0 & 0 & \alpha_2 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} + \begin{bmatrix} e_2 \\ f_2 \\ o_2 \end{bmatrix} \right) + \begin{bmatrix} e_1 \\ f_1 \\ 0 \end{bmatrix} = \begin{bmatrix} a_2 & b_2 & 0 \\ c_2 & d_2 & 0 \\ 0 & 0 & \alpha_1 \cdot \alpha_2 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} + \begin{bmatrix} e_1 + e_2 \\ f_1 + f_2 \\ o_2 \end{bmatrix}. \quad (3)$$

That is

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} a_2 & b_2 & 0 \\ c_2 & d_2 & 0 \\ 0 & 0 & \alpha_3 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} + \begin{bmatrix} e_1 + e_2 \\ f_1 + f_2 \\ o_2 \end{bmatrix}, \quad (4)$$

where  $\alpha_3 = \alpha_1 \cdot \alpha_2$ , which satisfies  $|\alpha_3| < 1$ .



**Fig. 6** (a) MSSBM that is eventually contractive and (b) typical cyclic transforms.



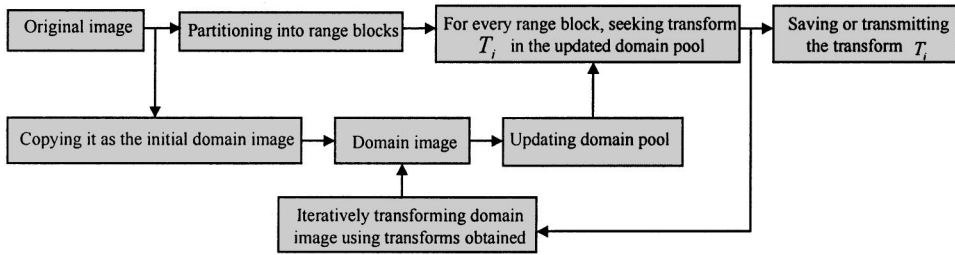


Fig. 7 The recursive coding method proposed.

This transform is obviously a contractive transform. It means that even though  $T_1$  is not a contractive transform, the compound transform  $T=T_1 \circ T_2$  is contractive.

In compound transform  $T=T_1 \circ T_2$ ,  $T_1$  is a same-sized mapping, and  $T_2$  is a twice-sized mapping. Since there is only one same-sized block mapping in the compound transform, we call this case single same-sized block mapping (SSSBM). SSSBM can exploit the same scale similarity to a degree, however, we can extend the idea to fully exploit the similarity at the same scale. We can construct a compound transform as  $T=T_1 \circ T_2 \circ \dots \circ T_k$ , where  $T_1, T_2, \dots, T_{k-1}$  are all same-sized block mappings, and only  $T_k$  is a contractive transform. It is easy to show that  $T$  is also contractive. We call such a case multiple same-sized block mapping (MSSBM). In applying MSSBM, we can encounter two cases illustrated in Fig. 6.

In Fig. 6(a), a block is encoded with a same-sized block  $B_2$ , part of  $B_2$  is encoded with another same-sized block  $B_3$ , and  $B_3$  is encoded with a same-sized block  $B_4$ , and

last,  $B_4$  is encoded with a twice larger domain block  $B_5$ . From the previous analysis, the compound transform is still contractive.

Another case in MSSBM is shown in Fig. 6(b). In this case, the same-sized block mappings  $T_1, T_2, T_3$ , and  $T_4$  form a cycle, i.e., block  $B_1$  is mapped with a same-sized block  $B_2$ , block  $B_2$  is mapped from a same-sized block  $B_3$ ,  $B_3$  is mapped with a same-sized block  $B_4$ , and  $B_4$  is mapped with  $B_1$ . In this case, the compound transform is not contractive like the case shown in Fig. 2. If a same-sized block mapping forms a cycle with other same-sized block mappings, we call it a cyclic transform.

In the practical application of MSSBM, when a same-size block mapping is found, we check if the mapping is cyclic to decide the usage.

The maximum number of possible same-sized block mapping=number of range blocks-1. On the average, the range blocks encoded with same-sized block mapping occupy 40% of all the range blocks. Only the position infor-

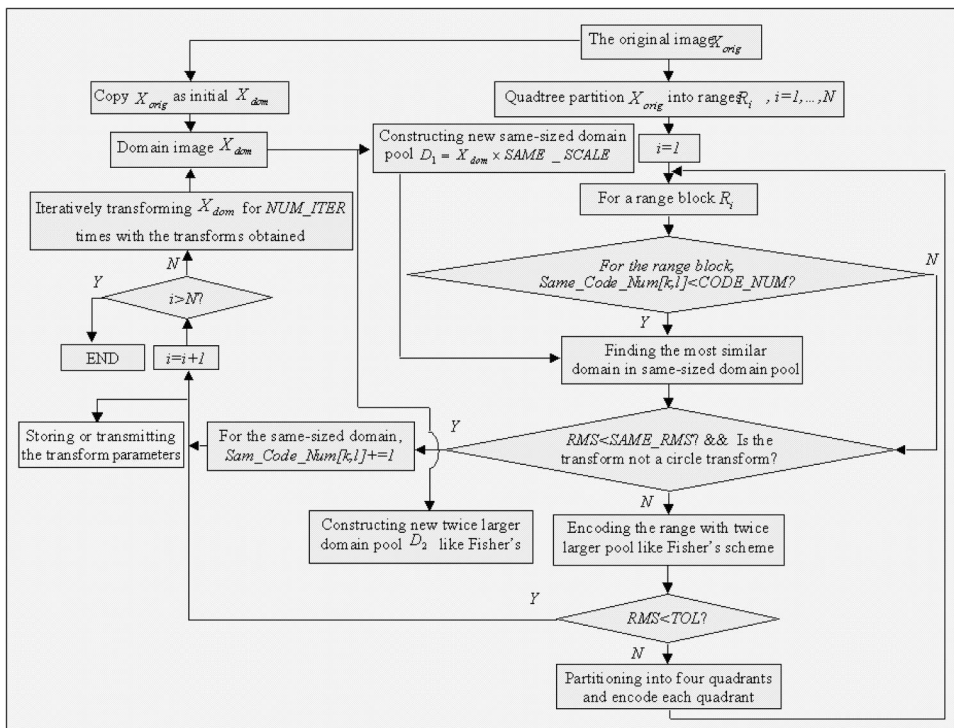
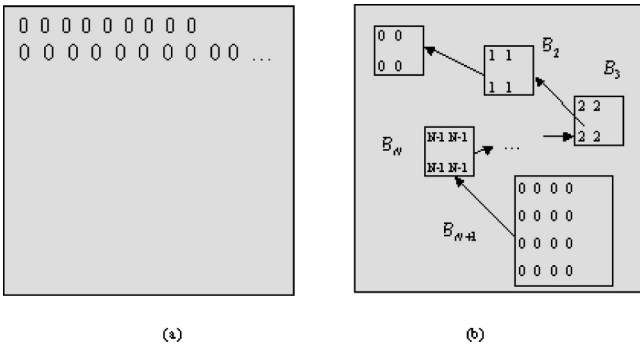


Fig. 8 The coding diagram of our method.



**Fig. 9** Illustration of the value changes of array Same\_Code\_Num  $[k, l]$ : (a) the values of all points of Same\_Code\_Num  $[k, l]$  are initialized as 0; and (b) the array values change as MSSBM proceeds.

mation of a same-sized domain needs storing. So, using MSSBM can significantly decrease the bitrate. Of course, an overhead of 1 bit is needed to distinguish the domain block types, resulting in a slightly decreased performance.

**3 Recursive Coding Scheme**

As discussed in Sec. 1, in common FIC encoding only the error between the image and its collage is minimized. However, the difference minimization between the image and its collage usually does not mean the minimization of the difference between the image and the decoded image.

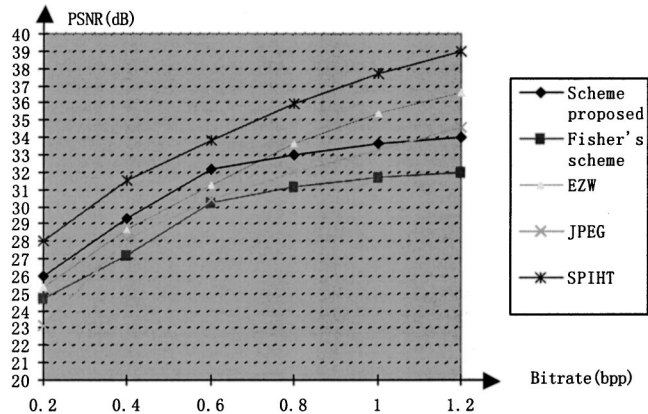
In fact, in the encoding, we should minimize the difference between the decoded image and the original. However, this is impossible by definition, since only after the image is encoded are the transforms obtained, and only after the transforms are obtained can the decoded image be reconstructed.

But, in the FIC scheme, the original image is encoded one range block after another, so in the encoding procedure, we obtain affine mappings for each range block in succession, therefore we can achieve an approximate decoded image (we call it domain image later) by iteratively transforming the original image using the transforms received. As the encoding procedure proceeds, the domain image is closer and closer to the decoded image. So we can use the domain image, which gradually reaches the decoded image, to construct the domain blocks pool and therefore gradually minimizes the error between the original image and the decoded image. The recursive coding method is shown in Fig. 7.

In this diagram, we partition the original image into range blocks and copy it as an initial domain image. For a range block, we construct its domain pool from the domain image, and search for an affine transform and a similar domain block in the domain pool. After that, we save or transmit its parameters. At the same time, we feed back the information and iteratively transform the old domain image using all the transforms obtained. Therefore we get a new domain image, which can be used for encoding the next range block. The range blocks are encoded one by one along with the domain image changing.

**4 Implementation of Our Scheme**

The coding steps of our scheme combining MSSBM and the recursive coding are shown in Fig. 8. As discussed in



**Fig. 10** PSNRs of the five methods versus different bitrates.

Sec. 2, a MSSBM can be considered as a compound transform  $T=T_1 \circ T_2 \circ \dots \circ T_K$ , where  $T_1, T_2, \dots, T_{K-1}$  are all same-sized block mappings, only  $T_K$  is a contractive transform. In our implementation diagram, CODE\_NUM denotes the preselected maximum number of same-sized block mappings in a compound transform, i.e.,  $K-1$ . Accordingly, Same\_Code\_Num  $[k, l]$  is a data array to denote the coding number of MSSBM in position  $(k, l)$ , the array is same-sized as the original image and initialized as 0. When a block  $B_1$  is transformed with a same-sized block  $B_2$ , the value of Same\_Code\_Num  $[k, l]$  in the area of  $B_2$  change to 1. When a block  $B_2$  is transformed with a same-sized block  $B_3$ , the value of Same\_Code\_Num  $[k, l]$  in the area of  $B_3$  changes to 2, shown in Fig. 9.

The term  $X_{orig}$  means the original image to be encoded, and  $X_{dom}$  is the domain image that is updated in the encoding and used to construct domain pools. In the beginning,  $X_{dom}$  is a copy of  $X_{orig}$ . SAME\_SCALE is a preselected multiplication scale for constructing the same-sized domain pool, and it can be in the range  $[0, 1]$ . The root-mean-square (RMS) error is between the range and its collage. SAME\_RMS is a preselected threshold for RMS in same-sized block mapping, such as 8, TOL is a preselected threshold for RMS in twice larger domain coding. NUM\_ITER is times, which we iteratively transform the domain image for.

**5 Experimental Results**

In the following experiments,  $256 \times 256 \times 8$  ‘‘Lena’’ is used as the test image.

**5.1 Experiment 1**

This experiment is conducted to compare the performance of the method proposed with some other well-known methods including Fisher’s quadtree scheme,<sup>3</sup> JPEG,<sup>16</sup> EZW,<sup>16</sup> and SPIHT.<sup>16</sup> The curves of the peak-to-peak signal to noise ratios (PSNR) versus bitrate (bpp, bits per pixel) in Fig. 10 summarize the results.

Analyzing the curves in Fig. 10, we know that at the same bitrate, the PSNR of our method is about 2 dB higher than that of Fisher’s, and the performance is also better than JPEG and EZW at a lower bitrate. Comparatively SPIHT is still the best scheme.

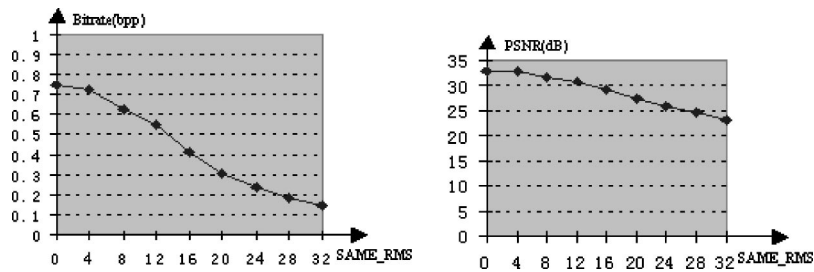


Fig. 11 The bitrate and PSNR versus (SAME\_RMS).

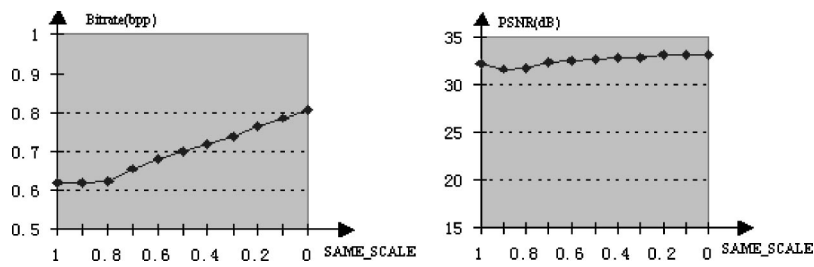


Fig. 12 Bitrate and PSNR versus (SAME\_SCALE).

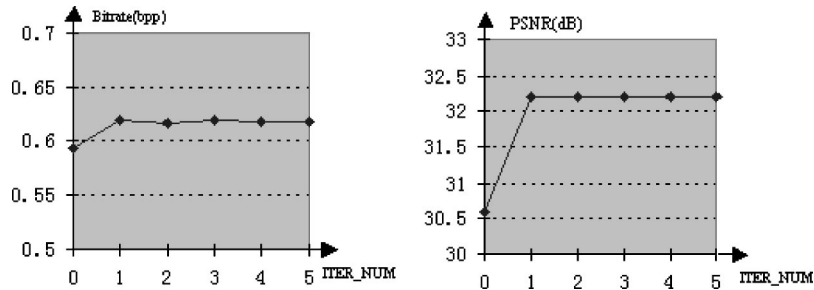


Fig. 13 Bitrate and PSNR versus (ITER\_NUM).

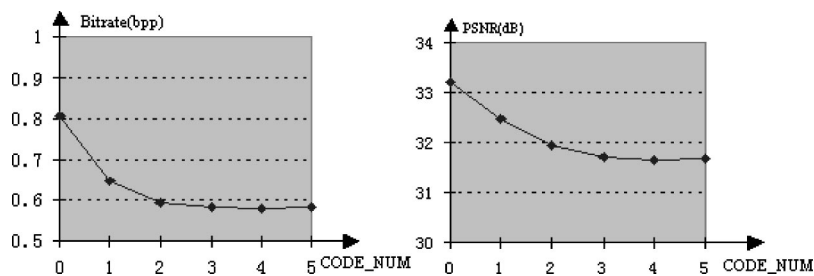


Fig. 14 Bitrate and PSNR versus (CODE\_NUM).

## 5.2 Experiment 2

The experiment is used to test the performance with the variation of parameters. Figure 11 shows the curves of bitrate and PSNR versus SAME\_RMS encoded with TOL = 8, CODE\_NUM = 5, and ITER\_NUM = 6. SAME\_RMS is the error threshold for same-sized block mapping. As SAME\_RMS increases, the bitrate decreases fast while PSNR decreases little.

Figure 12 shows the curves of bitrate and PSNR versus SAME\_SCALE encoded with TOL = 8, CODE\_NUM = 5, ITER\_NUM = 6, and SAME\_RMS = 8. From Fig. 12, we know that SAME\_SCALE = 0.8 to 1.0 can achieve good performance.

Figure 13 shows the curves of bitrate and PSNR versus ITER\_NUM encoded with TOL = 8, CODE\_NUM = 5, SAME\_RMS = 8, and SAME\_SCALE = 1.0. From Fig. 13, we can easily conclude that the decoded image quality with ITER\_NUM = 1 is much better than ITER\_NUM = 0, even though bpp increases a little. However, when ITER\_NUM increases from 1, the performance almost remains unchanged.

Figure 14 shows the curves of bitrate and PSNR versus CODE\_NUM encoded with TOL = 8, ITER\_NUM = 5, SAME\_RMS = 10, and SAME\_SCALE = 1. As CODE\_NUM increases, the bitrate decreases fast.

## 6 Conclusions

We propose a new scheme combining multiple same-sized block mapping and a recursive coding scheme. The scheme not only efficiently exploits the same scale similarity of the image, but also can timely feed the coding results back to modify coding parameters. Experimental results indicate that even though the performance of the scheme is still inferior to the SPIHT scheme of wavelets, it is an efficient improvement from conventional FIC.

### Acknowledgments

This work is supported in part by National Natural Science Foundation of China (Grant No. 69802001 and No. 60172062), Ministry of Education and Paper Foundation of NJTU. The authors would like to thank Professor Reginald L. Lagendijk and the anonymous referees for their help to improve the paper through constructive comments and suggestions.

### References

1. M. F. Barnsley and A. D. Sloan, "Chaotic compression," *Computer Graphics World*, 107–108 (Nov. 1987).
2. A. E. Jacquin, "A novel fractal block-coding technique for digital images," *Proc. ICASSP* 4, 2225–2228 (1990).
3. Y. Fisher, E. W. Jacobs, and R. D. Boss, "Fractal image compression using iterated transforms," in *Image and Text Compression*, James A. Storer, Ed., pp. 35–61, Kluwer Academic Publishers, Boston, MA (1992).
4. F. Davoine and J.-M. Chassery, "Adaptive Delaunay triangulation for attractor image coding," in *Proc. 12th Intl. Conf. Patt. Recog.* 801–803, IEEE Computer Society Press (1994).
5. F. Davoine, J. Svensson, and J.-M. Chassery, "A mixed triangular and quadrilateral partition for fractal image coding," *Proc. ICIP-95* 3, 284–287 (1995).

6. L. Thomas and F. Deravi, "Region-based fractal image compression using heuristic search," *IEEE Trans. Image Process.* 4(6), 832–838 (1995).
7. G. M. Davis, "A wavelet-based analysis of fractal image compression," *IEEE Trans. Image Process.* 7(2), 141–154 (1988).
8. Y. Zhao and B. Yuan, "Image compression using fractals and discrete cosine transform," *Electron. Lett.* 30(6), 474–475 (1994).
9. K. Kim and R.-H. Park, "Still image coding based on vector quantization and fractal approximation," *IEEE Trans. Image Process.* 5(4), 587–597 (1996).
10. D. M. Monro and F. Dudbridge, "Fractal block coding of images," *Electron. Lett.* 28(11), 1053–1055 (1992).
11. Y. Zhao and B. Yuan, "A new affine transformation: Its theory and application to image coding," *IEEE Trans. Circuits Syst.* 8(3), 269–274 (1998).
12. N. Lu, *Fractal Imaging*, Academic Press, New York (1997).
13. R. Hamzaoui, H. Hartenstein, and D. Saupe, "Local iterative improvement of fractal image codes," *Image Vis. Comput.* 18, 565–568 (2000).
14. H.-S. Kang and J.-W. Chuang, "Suboptimal fractal coding scheme using iterative transformation," *Opt. Eng.* 40(5), 703–712 (2001).
15. T. Bedford, F. M. Dekking, M. Breeuwer, M. S. Keane, and D. van Schooneveld, "Fractal coding of monochrome images," *Signal Process. Image Commun.* 6, 405–419 (1994).
16. See <http://www-it.et.tudelft.nl/~inald/vcdemo/index.htm>.



**Yao Zhao** received the BE degree from Fuzhou University in 1989, and ME degree from Southeast University in 1992, both from the radio engineering department. He received his PhD degree from the Institute of Information Science, Northern Jiaotong University (NJTU) in 1996. He became an associate professor with NJTU in 1998. He is an editor for a national journal *Signal Acquisition and Processing*. Now he has published more than 40 papers in international journals and conferences. He has led and been involved in several national research projects. His research interest includes image coding, fractals, and digital watermarking. He is now working as a senior Research Fellow in the Information and Communication Theory Group, Faculty of Information Technology and Systems, Delft University of Technology, The Netherlands.



**Hongxing Wang** is a senior member of the Chinese Institute of Electronics. He has published more than 30 papers. His research interest includes digital communication and digital image processing.



**Baozong Yuan** joined Northern Jiaotong University in 1953. He received his PhD degree in electrical engineering from the Leningrad Institute of Railway Engineering, USSR, in 1960. From 1982 to 1983, he was a visiting professor at the Department of Electrical and Computer Engineering in the University of Cincinnati, Ohio, and at the Department of Electrical Engineering in the University of Pittsburgh, Pennsylvania. He is an IEEE member, Chairman of the Computer Chapter of the IEEE Beijing Section, an IEE Fellow, and Vice Chairman of the IEE Beijing Center of Development. His research interests include digital signal processing, speech signal processing, image processing, computer vision, computer graphics, multimedia information processing, and data communication.