ELSEVIER

Signal Processing: *Image Communication* 15 (2000) 767–779

SIGNAL PROCESSING: IMAGE COMMUNICATION

www.elsevier.nl/locate/image

Hybrid image coding based on partial fractal mapping

Zhou Wang[a], David Zhang[b,*], Yinglin Yu[c]

[a]*Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, USA*
[b]*Department of Computing, Hong Kong Polytechnic University, Hong Kong*
[c]*Department of Electronic and Communication Engineering, South China University of Technology, Guangzhou, People's Republic of China*

Received 4 May 1998

Abstract

The fractal image compression technique models a natural image using a contractive mapping called fractal mapping in the image space. In this paper, we demonstrate that the fractal image coding algorithm is compatible with other image coding methods. In other words, we can encode only part of the image using fractal technique and model the remaining part using other algorithms. According to such an idea, a new mapping in the image space called partial fractal mapping is proposed. Furthermore, a general framework of fractal-based hybrid image coding encoding/decoding systems is presented. The framework provides us with much flexibility for real implementations. Many different hybrid image coding schemes can be derived from it. Finally, a new hybrid image coding scheme is proposed where non-fractal coded regions are used to help the encoding of fractal coded regions. Experiments show that the proposed system performs better than the quadtree-based fractal image coding algorithm and the JPEG image compression standard at high compression ratios larger than 30.

Keywords: Image coding; Fractal; Data compression; Hybrid system

$\mathscr{R}^M$	the M-dimensional Cartesian product of the real numbers
ϕ, ψ, θ	images (or elements in $\mathscr{R}^M$)
$d_{\max}$	the $d_{\max}$ metric of two elements in $\mathscr{R}^M$
d_2	the d_2 metric of two elements in $\mathscr{R}^M$
Ω	the set of sample point indexes in a signal
W	basic fractal mapping
W^{P}	partial fractal mapping

1. Introduction

The concept of fractal was introduced by Mandelbrot [8] as an alternative to the traditional Euclidean geometry mainly for dealing with shapes generated by nature. In recent years, the interest of applying this theory has been steadily growing. A recent trend in computer graphics and image processing has been to use iterated function system (IFS) to generate and describe both man-made fractal-like structures and natural images. Barnsley et al. were the first to present the concept of fractal image compression using IFS [1]. A fully automatic image compression algorithm for real-world gray scale images called fractal block coding (FBC) was proposed by Jacquin [4,5]. Jacquin's algorithm has been studied, refined and improved in recent years, many important research results on this topic are collected in Fisher's book [3]. As Jacquin indicated in [5], the main point of FBC is that it

* Corresponding author.

0923-5965/00/$ - see front matter
PII: S0923-5965(99)00018-1

can capture and exploit a special kind of image redundancy – piecewise self-similarity in images, which is not used by traditional image coding techniques. Since natural images are not exactly self-similar, the image will be formed of properly transformed 'parts' of itself [3]. The fractal code of the image is actually the collection of these transformations, which are called fractal transformations. Usually, fractal transformations can be represented by fewer bits than the original image, so a fractal code is a compression of the original image, sometimes with very high compression ratios.

Nevertheless, fractal representations are not always very efficient for image compression. For example, the fractal compression method usually needs much more bits to encode very 'flat' regions than other simpler methods. Taking advantage of the powerful compression ability of fractal coding while at the same time avoiding its inefficiency for some kinds of regions seems contradictory. A way to solve this problem is to develop a hybrid system where only some parts of the image are encoded using fractal techniques and the remaining parts are modeled using other methods, so that the merits of different approaches can be combined and the overall coding efficiency is improved. There have been some examples for such kind of hybrid systems. In Jacquin's primary work [4,5], he used only dc components to encode *shade blocks* (smooth with no significant gradient). Øien et al. reported that it might be advantageous to encode the complex regions (sharp edges and textures) of an image with FBC and smooth regions with a block-DCT encoder [5]. Laurencot and Jacquin compared FBC with LBG [7] based VQ scheme. Their results appeared that for sharp-edge blocks, FBC performs better than VQ, but the results are very similar for texture blocks, with perhaps a slight advantage for VQ coding of texture block [5,6].

It should be noted that all of the above-mentioned hybrid systems assume that the fractal image coding algorithm is compatible with other methods. However, no explanation on the compatibility had been given. Since in fractal block coding, the encoding of every block is highly related with other blocks in the image, if only a proportion of the blocks are coded using fractal method, could the remaining non-fractal-coded blocks destroy the whole fractal code?

In this paper, by introducing the concept of partial fractal mapping, we provide some mathematical explanations on how and why a hybrid fractal image coding system can work. A general framework of the hybrid system is then proposed. According to such a framework, many different hybrid image coding schemes can be derived. Actually, all the above-mentioned hybrid systems are only some special cases of it. The framework also gives us something that is new to those previously proposed algorithms: 1. Non-fractal-coded regions can be used to help the encoding of the fractal-coded regions to improve the overall coding efficiency; 2. Non-fractal-coded regions can be encoded using not only block-based but also non-block-based algorithms; 3. The domain pool used by the fractal-coded blocks can be reduced. Finally, we provide an example of the new hybrid system and give some experimental results.

2. Basic fractal mapping

A digital signal ϕ with M sample points can be viewed as an element in $\mathscr{R}^M$, where $\mathscr{R}^M$ denotes the M-dimensional Cartesian product of the real numbers. Let $\Omega = \{1, 2, \ldots, M\}$ represent the set of sample point indexes in the signal. We can denote $\phi \in \mathscr{R}^M$ and $\phi = (\phi(1), \phi(2), \ldots, \phi(M))^{\mathrm{T}}$. In this paper, the signal ϕ refers to an image and $\mathscr{R}^M$ refers to the image space that is the collection of all possible images. When we use sub-indexes on ϕ such as ϕ_1 and ϕ_n, we mean they are different images. For convenience, one-dimensional model is used in this paper. Nevertheless, the discussions and theorems presented below are also applicable or can be easily adapted to two-dimensional or higher-dimensional cases.

In $\mathscr{R}^M$, we define the following two kinds of metrics:

$$\forall \phi_1, \phi_2 \in \mathscr{R}^M, \quad d_{\max}(\phi_1, \phi_2) = \max_{j \in \Omega} |\phi_1(j) - \phi_2(j)|, \tag{1}$$

$\forall \phi_1, \phi_2 \in \mathscr{R}^M$,

$$d_2(\phi_1, \phi_2) = \left(\frac{1}{M} \sum_{j=1}^{M} |\phi_1(j) - \phi_2(j)|^2 \right)^{1/2}. \tag{2}$$

Then we have two complete metric spaces of $(\mathscr{R}^M, d_{\max})$ and $(\mathscr{R}^M, d_2)$. It is easy to prove

$$\forall \phi_1, \phi_2 \in \mathscr{R}^M, \quad d_2(\phi_1, \phi_2) \leqslant d_{\max}(\phi_1, \phi_2). \tag{3}$$

The image is partitioned into N non-overlapping sub-pieces $R_i \subset \Omega$ $(i = 1, 2, \ldots, N)$. In the fractal image coding literature, R_i is also called a range block. Let r_i be the number of pixels in R_i $(M = \sum_{i=1}^{N} r_i)$, then we have

$$\Omega = \bigcup_{i=1}^{N} R_i \quad \text{and} \quad R_i \cap R_j = \varnothing \quad (i \neq j). \tag{4}$$

Each R_i corresponds to a transformation W_i, which transforms another sub-piece in the image $D_i \subset \Omega$ to the position of R_i, where D_i is also called a domain block and we use d_i to represent the number of pixels in D_i. W_i is a combination of the extracting transformation E_i, the geometrical transformation G_i, the luminance transformation L_i and the putting transformation P_i. They are illustrated as follows:

- $E_i : \mathscr{R}^M \to \mathscr{R}^{d_i}$ is to extract the domain block D_i from the image.
- $G_i : \mathscr{R}^{d_i} \to \mathscr{R}^{r_i}$ includes the action of spatial contraction and rotation, that maps block of size D_i to block of size R_i and rotate it in some way. Usually d_i is a multiple of r_i, so that every pixel j in R_i corresponds to K pixels $j_1, j_2, \ldots, j_K$ in D_i. The pixel value of j is determined by the weighted average of corresponding pixel values in D_i, that is,

$$z_R(j) = \sum_{k=1}^{K} a_{j_k} \cdot z_D(j_k), \tag{5}$$

  where z_R and z_D are the gray values in R_i and D_i, respectively. $(a_{j_1}, a_{j_2}, \ldots, a_{j_K})$ is the weight vector that satisfies $\sum_{k=1}^{K} a_{j_k} = 1$.
- $L_i : \mathscr{R}^{r_i} \to \mathscr{R}^{r_i}$ is a luminance transformation which modifies each pixel one by one using a uniform one-dimensional gray value mapping $l_i : \mathscr{R} \to \mathscr{R} z \to l_i(z)$. If

$$\exists\, 0 \leqslant s < 1, \quad \forall i \in [1, N], \quad \forall z_1, z_2 \in \mathscr{R}$$
$$|l_i(z_1) - l_i(z_2)| < s \cdot |z_1 - z_2|, \tag{6}$$

  then l_i is a one-dimensional contractive mapping.
- $P_i : \mathscr{R}^{r_i} \to \mathscr{R}^M$ is to put the processed image block back into the image at the position of R_i and set the remaining part of the image to be all zero.

By applying the transformations in order of E_i, G_i, L_i and P_i, we can achieve the combined transformation of

$$W_i : \mathscr{R}^M \to \mathscr{R}^M, \quad W_i(\phi) = P_i L_i G_i E_i(\phi). \tag{7}$$

The collection of all W_i is called an iterated function system: $\mathrm{IFS} = \{W_1, W_2, \cdots, W_N\}$. The function of IFS can also be viewed as an overall fractal mapping W on the image, which is the sum of all W_i:

$$W : \mathscr{R}^M \to \mathscr{R}^M, \quad W(\phi) = \sum_{i=1}^{N} W_i(\phi). \tag{8}$$

If all l_i's are contractive, it can be proved that W is a contractive mapping in $(\mathscr{R}^M, d_{\max})$, that is,

$$\forall \phi_1, \phi_2 \in \mathscr{R}^M,$$
$$d_{\max}(W(\phi_1), W(\phi_2)) \leqslant s \cdot d_{\max}(\phi_1, \phi_2), \tag{9}$$

where $0 \leqslant s < 1$ is called the contractive factor of W. According to the Contraction Mapping Theorem in complete metric space, W has the following properties.

Property 1. There exists a unique attractor image $\phi \in \mathscr{R}^M$, such that

$$W(\phi) = \phi. \tag{10}$$

Property 2. Iteratively, apply W to any initial image $\phi^{(0)}$ in $\mathscr{R}^M$, the attractor image can be eventually obtained:

$$\forall \phi^{(0)} \in \mathscr{R}^M, \quad \lim_{n \to \infty} W^{\circ n}(\phi^{(0)}) = \phi, \tag{11}$$

where

$$\phi^{(n)} = W^{\circ n}(\phi^{(0)}) = \underbrace{W(W(\cdots(W}_{n \text{ times}}(\phi^{(0)}))).$$

Property 3. The collage theorem: $\forall \psi \in \mathscr{R}^M$, if for $\varepsilon > 0$, we have $d_{\max}(W(\psi), \psi) \leqslant \varepsilon$, then

$$d_{\max}(\phi, \psi) \leqslant \frac{\varepsilon}{1-s}. \tag{12}$$

These three properties are the bases of fractal image coding algorithms. If we have an image ψ to be encoded, then the encoding is to find a fractal mapping W, which makes the error between $W(\psi)$ and ψ as small as possible. When decoding, W is applied iteratively to an arbitrary initial image $\phi^{(0)}$, then we obtain an image sequence $\phi^{(0)}$, $\phi^{(1)}$, …, $\phi^{(n)}$, … and the attractor image ϕ is eventually achieved, which is our reconstructed image. According to the collage theorem, the difference between ϕ and ψ can be controlled by the error between $W(\psi)$ and ψ. In practice, even if some of the l_i's are non-contractive, the iteration procedure can still converge to an attractor image. Although such kind of convergence is very difficult to be mathematically proved, the usage of non-contractive l_i has been widely adopted by many fractal image coding systems. More discussions are given by Fisher in [2].

3. Partial fractal mapping

If only part of the image is coded using fractal method, then Eq. (4) should be modified. First, we partition the whole image Ω into two parts Ω_1 and Ω_2,

$$\Omega = \Omega_1 \cup \Omega_2 \quad \text{and} \quad \Omega_1 \cap \Omega_2 = \varnothing, \tag{13}$$

where only Ω_1 part is modeled using fractal method and Ω_2 part is encoded using any other method. Ω_1 can be further partitioned,

$$\Omega_1 = \bigcup_{i=1}^{N_1} R_i \quad \text{and} \quad R_i \cap R_j = \varnothing \quad (i \neq j). \tag{14}$$

Under such a partitioning, we have $\sum_{i=1}^{N_1} r_i < M$. Obviously, $M - \sum_{i=1}^{N_1} r_i$ is the number of pixels in Ω_2.

For the non-fractal coded part Ω_2, we use a constant function mapping $C : \mathscr{R}^M \to \mathscr{R}^M$, so that $\forall \phi \in \mathscr{R}^M$, we always have $C(\phi) = \theta$, where θ is a constant vector in $\mathscr{R}^M$ and satisfies $\forall j \in \Omega_1$, $\theta(j) = 0$.

For the fractal coded part Ω_1, the process is similar to that in Section 2.1 for basic fractal encoding. For each R_i, we find the appropriate W_i and D_i. As to a certain R_i, its corresponding D_i may in Ω_1 or in Ω_2, or part of it in Ω_1, and the remaining part in Ω_2.

We call the collection of W_i $(i = 1, \ldots, N_1)$ a partial iterated function system: $\mathrm{IFS}^{\mathrm{P}} = \{W_1, W_2, \ldots, W_{N_1}\}$. The combination of $\mathrm{IFS}^{\mathrm{P}}$ and the constant mapping C compose a partial fractal mapping,

$$W^{\mathrm{P}} : \mathscr{R}^M \to \mathscr{R}^M,$$

$$W^{\mathrm{P}}(\phi) = C(\phi) + \sum_{i=1}^{N_1} W_i(\phi) = \theta + \sum_{i=1}^{N_1} W_i(\phi). \tag{15}$$

Now, we discuss the properties of W^{P}.

Theorem 1. *W^{P} is a contractive mapping in $(\mathscr{R}^M, d_{\max})$.*

Proof. $\forall \phi_1, \phi_2 \in \mathscr{R}^M$ $\forall j \in \Omega$

if $j \in \mathscr{R}_i \subset \Omega_1$, then

$$\begin{aligned}
&|(W^{\mathrm{P}}\phi_1)(j) - (W^{\mathrm{P}}\phi_2)(j)| \\
&= \left| v_i\left(\sum_{k=1}^{K} a_{j_k}\phi_1(j_k)\right) - v_i\left(\sum_{k=1}^{K} a_{j_k}\phi_2(j_k)\right)\right| \\
&\overset{(6)}{<} s \cdot \left|\sum_{k=1}^{K} a_{j_k}\phi_1(j_k) - \sum_{k=1}^{K} a_{j_k}\phi_2(j_k)\right| \\
&= s \cdot \left|\sum_{k=1}^{K} a_{j_k}[\phi_1(j_k) - \phi_2(j_k)]\right| \\
&\leqslant s \cdot \sum_{k=1}^{K} a_{j_k}|\phi_1(j_k) - \phi_2(j_k)| \\
&\leqslant s \cdot \sum_{k=1}^{K} a_{j_k} \cdot \max_{j=1,\ldots,M} |\phi_1(j) - \phi_2(j)| \\
&= s \cdot \max_{j=1,\ldots,M} |\phi_1(j) - \phi_2(j)| = s \cdot d_{\max}(\phi_1, \phi_2),
\end{aligned} \tag{16}$$

if $j \in \Omega_2$, then

$$|(W^{\mathrm{P}}\phi_1)(j) - (W^{\mathrm{P}}\phi_2)(j)| = |\theta(j) - \theta(j)| = 0 \leqslant s \cdot d_{\max}(\phi_1, \phi_2), \tag{17}$$

therefore

$$\max_{j=1,\ldots,M} |(W^{\mathrm{P}}\phi_1)(j) - (W^{\mathrm{P}}\phi_2)(j)| \leqslant s \cdot d_{\max}(\phi_1, \phi_2), \tag{18}$$

i.e.,

$$d_{\max}(W^{\mathrm{P}}(\phi_1), W^{\mathrm{P}}(\phi_2)) \leqslant s \cdot d_{\max}(\phi_1, \phi_2). \quad \square \tag{19}$$

According to Theorem 1 and the Contractive Mapping Theorem in complete metric space, we can prove that in $(\mathscr{R}^M, d_{\max})$ space, W^{P} satisfies the three properties of W described in Section 2.1.

In real applications, we are more concerned with the d_2 metric. However, in $(\mathscr{R}^M, d_2)$, it is difficult to prove the contractivity of either W or W^{P}, but we have the following theorems.

Theorem 2. *There are a basic fractal mapping W and a partial fractal mapping W^{P} in $(\mathscr{R}^M, d_2)$ space, where the* IFS$^{\mathrm{P}}$ *of W^{P} is a subset of the IFS of W* (i.e., *the encoding of Ω_1 part of W^{P} is exactly the same as the corresponding part of W). If W is a contractive mapping, then W^{P} is also a contractive mapping.*

Proof. $\forall \phi_1, \phi_2 \in \mathscr{R}^M$
Because W is contractive in $(\mathscr{R}^M, d_2)$, $d_2(W(\phi_1), W(\phi_2)) \leqslant s \cdot d_2(\phi_1, \phi_2)$. Thus, we have

$$\begin{aligned} &d_2(W^{\mathrm{P}}(\phi_1), W^{\mathrm{P}}(\phi_2)) \\ &= \left(\frac{1}{M}\sum_{j=1}^{M} |(W^{\mathrm{P}}\phi_1)(j) - (W^{\mathrm{P}}\phi_2)(j)|^2\right)^{1/2} \\ &= \left(\frac{1}{M}\sum_{j\in\Omega_1} |(W\phi_1)(j) - (W\phi_2)(j)|^2 + \frac{1}{M}\sum_{j\in\Omega_2} |\theta(j) - \theta(j)|^2\right)^{1/2} \\ &\leqslant \left(\frac{1}{M}\sum_{j=1}^{M} |(W\phi_1)(j) - (W\phi_2)(j)|^2\right)^{1/2} \\ &= d_2(W(\phi_1), W(\phi_2)) \leqslant s \cdot d_2(\phi_1, \phi_2) \quad \square \end{aligned} \tag{20}$$

Theorem 2 demonstrates that W^{P} is more likely to be contractive than W in $(\mathscr{R}^M, d_2)$ space.

Theorem 3. *In $(\mathscr{R}^M, d_2)$ space, W^{P} has a unique attractor image ϕ, such that $W^{\mathrm{P}}(\phi) = \phi$*

Proof. (*Existence.*) From the properties of W^{P} in $(\mathscr{R}^M, d_{\max})$ space, we know that there exists a $\phi \in \mathscr{R}^M$ that makes $d_{\max}(W^{\mathrm{P}}(\phi), \phi) = 0$, then we have

$$d_2(W^{\mathrm{P}}(\phi), \phi) \overset{(3)}{\leqslant} d_{\max}(W^{\mathrm{P}}(\phi), \phi) = 0. \tag{21}$$

Therefore in $(\mathscr{R}^M, d_2)$ space, $W^{\mathrm{P}}(\phi) = \phi$.

That is to say, W^{P} has the same attractor image ϕ in both $(\mathscr{R}^M, d_{\max})$ and $(\mathscr{R}^M, d_2)$.

(*Uniqueness.*) If there exist two different attractors ϕ_1, ϕ_2 of W^{P} in $(\mathscr{R}^M, d_2)$ space,

$$\phi_1 \neq \phi_2, \quad W^{\mathrm{P}}(\phi_1) = \phi_1 \quad \text{and} \quad W^{\mathrm{P}}(\phi_2) = \phi_2, \tag{22}$$

then

$$[d_2(W^{\mathrm{P}}(\phi_1), \phi_1)]^2 = \frac{1}{M}\sum_{j=1}^{M} |(W^{\mathrm{P}}\phi_1)(j) - \phi_1(j)|^2 = 0. \tag{23}$$

Obviously, $\forall j \in \Omega, |(W^{\mathrm{P}}\phi_1)(j) - \phi_1(j)| = 0$

so, $\max_{j=1,\ldots,M} |(W^{\mathrm{P}}\phi_1)(j) - \phi_1(j)| = 0$,

i.e., $d_{\max}(W^{\mathrm{P}}(\phi_1), \phi_1) = 0.$ (24)

Therefore ϕ_1 is an attractor of W^{P} in $(\mathscr{R}^M, d_{\max})$ space.

For the same reason, ϕ_2 is also an attractor in $(\mathscr{R}^M, d_{\max})$ space.

But we have assumed $\phi_1 \neq \phi_2$. This is contradictory to the theorem that the attractor of W^{P} in $(\mathscr{R}^M, d_{\max})$ space is unique. $\square$

Theorem 4. *If ϕ is the attractor of W^{P} in $(\mathscr{R}^M, d_2)$ space, then*

$$\forall \phi^{(0)} \in \mathscr{R}^M, \quad \lim_{n\to\infty} (W^{\mathrm{P}})^{\circ n}(\phi^{(0)}) = \phi. \tag{25}$$

Proof. According to Property 2 of W^{P} in $(\mathscr{R}^M, d_{\max})$ space,

$$\forall \varepsilon > 0, \exists N, \forall n > N, \quad d_{\max}((W^{\mathrm{P}})^{\circ n}(\phi^{(0)}), \phi) < \varepsilon \tag{26}$$

$$\text{so,} \quad d_2((W^{\mathrm{P}})^{\circ n}(\phi^{(0)}), \phi) \overset{(3)}{\leqslant} d_{\max}((W^{\mathrm{P}})^{\circ n}(\phi^{(0)}), \phi) < \varepsilon, \tag{27}$$

therefore

$$\lim_{n \to \infty} (W^{\mathrm{P}})^{\circ n}(\phi^{(0)}) = \phi. \quad \square \tag{28}$$

Theorems 3 and 4 prove that W^{P} still satisfies Properties 1 and 2 in $(\mathscr{R}^M, d_2)$ space. The difference from those in $(\mathscr{R}^M, d_{\max})$ space is that we cannot prove the collage theorem (Property 3). However, the error between the original image and the reconstructed image still can be controlled: $\forall \psi \in \mathscr{R}^M$, if $\exists \varepsilon > 0$, such that $d_{\max}(W^{\mathrm{P}}(\psi), \psi) \leqslant \varepsilon$, then we have

$$d_2(\phi, \psi) \overset{(3)}{\leqslant} d_{\max}(\phi, \psi) \leqslant \frac{\varepsilon}{1 - s}. \tag{29}$$

4. A general framework for hybrid fractal-based image coding systems

According to the construction and properties of W^{P}, we get a general hybrid fractal image coding framework which is shown in Fig. 1. First, an image partitioning algorithm segment the whole image into two parts Ω_1 and Ω_2. Ω_2 is encoded using some other techniques. Its decoding result is the item of θ in Eq. (15). The Ω_1 part is encoded using fractal method. The output codes include the fractal code of Ω_1 (a), the code of Ω_2 part (b) and the segmentation information code (c). When decoding, the segmentation information is first decoded, then we decode Ω_2 and get θ. Finally, the fractal code is iteratively applied to reconstruct part Ω_1 and we achieve our reconstructed image.

The framework of Fig. 1 provides us with much flexibility for real implementation. The flexibility exists in almost every section of the encoding side:

1. The image partitioning algorithm.
2. The encoding of the image partitioning information. Obviously, this information must be encoded using lossless methods.
3. The encoding of the non-fractal-coded region Ω_2. Not just block-based methods can be used as those in previous hybrid image coding systems. Actually, almost any image coding techniques can be chosen.

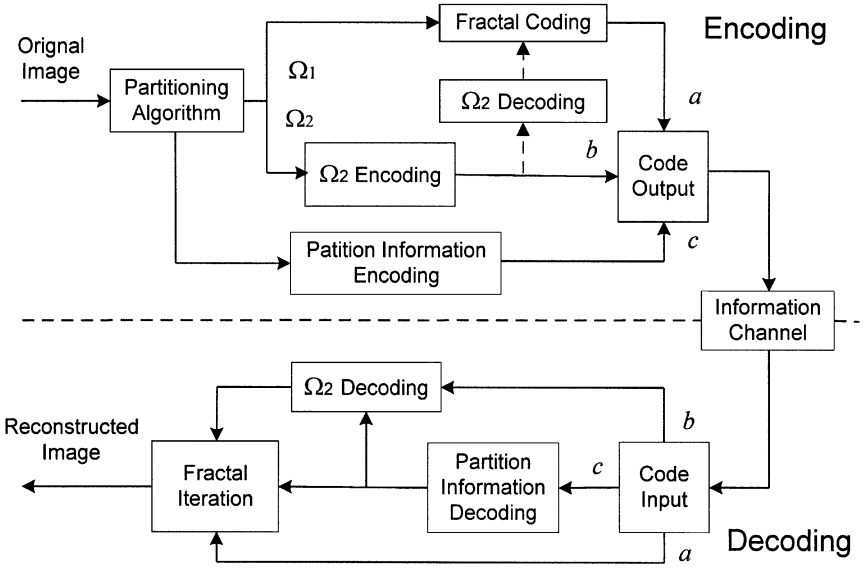

Fig. 1. A general framework of the hybrid image encoding/decoding systems.

4. The fractal image coding of Ω_1, which includes the selection of parameters, domain pool and the searching procedure, etc.
5. Sometimes the decoding result of Ω_2 can be used to predict some fractal parameters for the encoding of Ω_1, so that the coding efficiency of Ω_1 can be improved. We show this in Fig. 1 using the dashed lines.

Apparently, many different hybrid image coding schemes can be derived from the general framework shown in Fig. 1. Examples include those published in [4,6,7]. In the next section, we give another example which shows more merits of our hybrid method.

5. A hybrid image coding system

The image to be encoded is partitioned into non-overlapping equal-sized blocks. Each block is categorized into one of the following four classes:

1. A large smooth region which is composed of at least two of the connected adjacent blocks and can be approximated sufficiently well by a uniformly gray region.
2. A large smooth region which is composed of at least two of the connected adjacent blocks and can be sufficiently approximated by a plane (i.e., linear approximation).
3. An isolatedly coded block simply approximated by a uniformly gray block.
4. An isolatedly coded block modeled by fractal-based method.

Two bits are needed for each block to catalogue it into one of the four classes. We use a lossless adaptive arithmetic coding algorithm presented in [12] to further compress this information.

5.1. Image partitioning

First, a *Sobel* operator (refer to [13]) is applied to the original image and we obtain an edge image. An example of the edge image of 'Lena' is shown in Fig. 2. Second, the edge image is partitioned into non-overlapping equal-sized 8×8 blocks. An 8×8 block may have many edge pixels or only a few or no edge pixels. A threshold T_S of the number of edge pixels is used to determine whether the block is smooth or complex. Next, we use a region growing algorithm to merge adjacent smooth blocks (A block's top, right, bottom and left blocks are called its adjacent blocks). The region growing is an iterative procedure. We first select a smooth block from the image. If any of its four adjacent blocks is also smooth, then the middle block and the adjacent blocks are merged into a smooth region and the merged adjacent block becomes a new selected smooth block. The same merging process is applied to each of the new selected smooth blocks. The iteration continues so that the merged region grows as much as possible with at least two or sometimes hundreds of complete 8×8 blocks. For the ease of encoding, we restrict each merged region to be within a local 64×64 block region. An example of image partitioning and merging result for the standard image 'Lena' is shown in Fig. 3.

Fig. 2. Edge image of 'Lena'.

Except for the large smooth regions, the remaining parts of the image are all isolated blocks which will be encoded using block-based algorithms. These blocks are likely to be complex, but some of them may be smooth because a smooth block with no adjacent smooth block cannot be merged into a large smooth region.

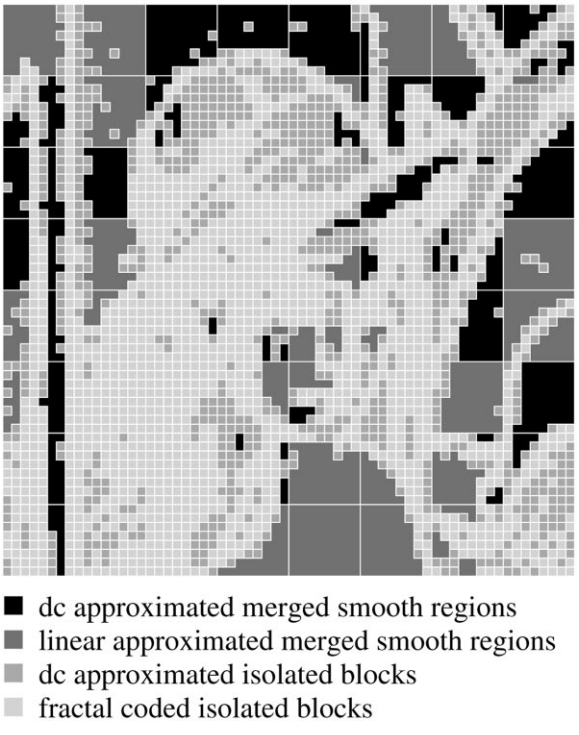

Fig. 3. An example of image partitioning and merging result for 'Lena' 512×512.

5.2. Approximation of merged smooth regions

For each merged smooth region, we first simply try to approximate it by a uniformly gray region with gray level equal to its mean value m. If the resulting MSE is less than a threshold T_{E1}, the dc component of the region is quantized and stored. Otherwise, we use a plane to approximate it. Plane approximation means to use a linear equation $\hat{z} = ax + by + c$ to model the region. The optimal parameters can be easily obtained under the condition of minimum squared error between the approximation plane and the original region. We can also easily prove that

$$c = m - a\bar{x} - b\bar{y}, \tag{30}$$

where m is the mean value (dc component) of the region, $\bar{x}$ and $\bar{y}$ are the averages of x and y within the region, respectively. Then the plane equation can be rewritten as

$$\hat{z} = ax + by + c = a(x - \bar{x}) + b(y - \bar{y}) + m. \tag{31}$$

We encode a, b and m to represent the plane.

5.3. Prediction of dc components of isolated blocks

The dc component of an isolated block is correlative with its neighborhood regions. We use the average of the dc components of its eight surrounding blocks as a prediction of the dc component of the surrounded block. If all isolated blocks have all of their surrounding blocks in merged smooth regions, the prediction is simple. However, an isolated block may have less than eight surrounding blocks in merged smooth regions, thus its dc component cannot be directly predicted. We use a progressive procedure to solve this problem. In the initial image, the merged smooth regions are covered using the approximation method described in Section 5.2 and the isolated blocks are left uncovered. The procedure consists of several steps. The output of each step is used as the input of the next step. In Step 1, for each isolated block, if all of its eight surrounding blocks have been covered, we compute the average of the dc components of these eight blocks as a prediction of its dc component. The difference between the real dc component and the predicted dc component is stored. Then the block is covered with a uniformly gray value equal to its real dc component. Steps 2,3,4, ... are the same as Step 1 except that the uncovered blocks are allowed to be covered if they have 7, 6, 5, ... instead of eight surrounding blocks covered. This procedure continues until all isolated blocks have been covered.

Normally, the difference between the real dc component and the predicted dc component is a value near zero. This kind of distribution allows us to code dc components more efficiently. If the absolute value of the difference between the real and predicted dc components is less than 16, we use 5 bits to code the difference. Otherwise, we use 8 bits to directly encode the real dc component. One extra bit is used to make a classification. If p represents the proportion of the 5 bits coded dc components in all dc components, then the average number of bits needed to code a dc component is

$5p + 8(1 - p) + 1 = 9 - 3p$. Since the dc components are useful for the encoding of isolated blocks including fractal-coded blocks, this prediction procedure can be viewed as a way where non-fractal-coded regions are used to help the encoding of fractal-coded regions.

5.4. Encoding of isolated blocks

For each isolated block, we also first simply try to approximate it by a uniformly gray block with gray level equal to its dc component. If the resulting MSE is less than a threshold T_{E2}, we do not need any extra bit to code it because its dc component has been predicted and quantized using the method described in Section 5.3. Otherwise, we encode it using a modified fractal block coding (FBC) method called fractal block coding in residue domain (FBCRD). A detailed description of FBCRD is presented in [11]. FBCRD is, in some ways, similar to a fractal image coding algorithm proposed by Øien and Lepsøy [9]. Both of them can get faster decoding than basic FBC and store the scaling factor and the dc component instead of the scaling factor and the offset for luminance transformations. This is beneficial with respect to encoding. FBCRD also has something different from Øien and Lepsøy's algorithm. When decoding, the dc component of the domain block is dynamically computed and is variable in each iteration, while in Øien and Lepsøy's algorithm, it is a fixed value. This difference frees FBCRD from several constraints on Øien and Lepsøy's algorithm:

1. There is no constraint on the image partitioning method.
2. It is not necessary for the domain block to be made up of an integer number of complete range blocks.
3. More important for our usage, FBCRD can be applied in a hybrid system where only part of the image is fractal-coded. In this kind of systems, since a domain block may cover some non-fractal coded regions, the dc component of the block can only be computed dynamically.

In our hybrid system, all fractal-coded blocks are complex blocks, thus more likely to be approximated by complex domain blocks. This allows us to exclude smooth domain blocks from the domain pool. If a domain block does not contain any pixel of any fractal-coded block, we regard it as a smooth domain block and exclude it from the domain pool. The reduction of domain pool leads to two improvements with respect to encoding. First, smaller domain pool means less encoding time. Second, smaller domain pool allows us to use fewer bits to encode the position shift for the geometrical transformations.

5.5. Decoding and postprocessing

At the decoder, the block classification information is the first to be decoded. Then large smooth regions are merged and covered using dc or linear approximation method. A progressive procedure corresponding to that in the encoder is applied to decode the dc components of isolated blocks and cover them with uniformly gray blocks. The result of this progressive procedure is our initial image for fractal decoder iterations. Then FBCRD-based decoding procedure is iteratively applied and more detailed information in fractal coded blocks is gradually emerged through iterations. The iteration is terminated when we achieve a satisfactory image. In each fractal decoder iteration, non-fractal-coded regions are kept unchanged.

To reduce the discontinuities at the block and region boundaries, we use a simple smooth algorithm similar to that introduced by Fisher in [2] to minimize these artifacts. That is, if the pixel values on either side of a boundary are u and v, then they are replaced by $\frac{3}{4}u + \frac{1}{4}v$ and $\frac{1}{4}u + \frac{3}{4}v$, respectively. This algorithm is applied only to boundary pixels of large smooth regions and isolated blocks. The internal pixels of either large regions or isolated blocks are kept unaltered.

6. Simulations and conclusions

We use some 8 bits/pixel gray scale images to test our hybrid system. Peak signal-to-noise ratio (PSNR) is used to make an objective evaluation. In our hybrid system, each 8×8 block needs 2 bits to be classified into one of four classes. Usually, this

classification information can be compressed by 25–40% by using adaptive arithmetic coding algorithm [12]. We use 8 bits to code the dc component for a dc approximated large smooth region and 8 bits to code each of the 3 parameters for a linear approximated merged smooth region. For isolated blocks, the proportion p is typically around 0.65, so $9 - 3p \approx 7$ is the average number of bits for a dc component. For a fractal-coded block, we use 6 bits to code scaling factor and 3 bits to code symmetry. Since the domain pool is condensed, the bits needed for the position of domain block is reduced from 10 to 9 bits. Averagely, a total of about 25 bits are needed for each fractal-coded block including the bits for its dc component.

The selection of the three threshold parameters T_S, T_{E1} and T_{E2} can be used to control the tradeoff between high compression ratio and good image quality. If T_S is higher, more blocks will be regarded as smooth, thus we will obtain higher compression ratio but lose more image details. Similarly, if T_{E1} and T_{E2} are higher, more blocks or regions will be simply represented using a uniformly gray region and the compression will be higher, but approximation becomes less efficient. Fig. 4 shows PSNR versus compression ratios and compares our hybrid system with Fisher's quadtree fractal block coding method [2] and the JPEG still image coding standard [10]. It appears that our hybrid system performs better than the fully fractal-coded quadtree method in a wide range of compression ratios. JPEG is better at low compression ratios, but the hybrid system outperforms JPEG at high compression ratios larger than 30. Figs. 5 and 6 show the decoded images of 'Lena' and 'Peppers' by our hybrid system, the quadtree method and the JPEG standard at similar compression ratios. In the hybrid system, the threshold parameters are set to $T_S = 0$, $T_{E1} = 32$ and $T_{E2} = 64$, respectively. In Fig. 7, some enlarged areas of the decoded images by the hybrid system and the quadtree method are provided, where the areas are extracted from Fig. 5. Our hybrid system appears to show major improvements with respect to the subjective quality of the reconstructed images.

Another benefit from our hybrid system is the save of encoding and decoding computation time. The reason is manifold. First, the number of fractal coded blocks is reduced compared with fully fractal block coding algorithm, while the encoding of other regions is much faster. Second, the domain pool for fractal block coding is reduced. Since most of the encoding time of fractal image compression systems is used in searching for suitable domain blocks in the domain pool, smaller domain pool means less encoding time. Third, smaller domain pool allows us to use fewer bits to code the position shift for the geometrical transformations. Experimentally, only 20–60% the encoding time of the fully fractal block coding algorithm is needed. Finally, the number of decoder iterations is less than the fully fractal coding algorithm due to the good convergence property of FBCRD.

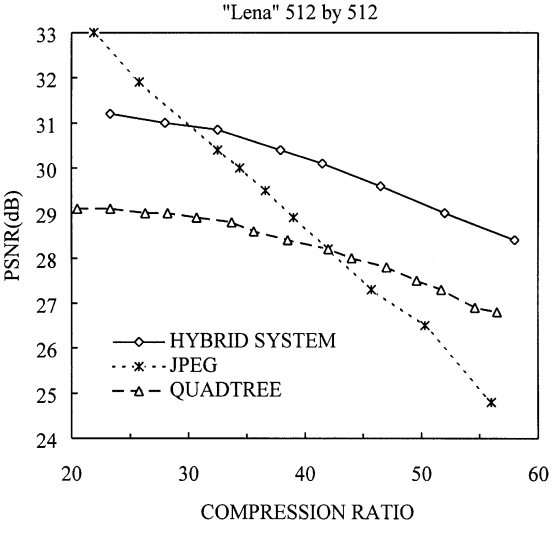

Fig. 4. A comparison of compression results for image 'Lena' 512×512.

In this paper, we propose the concept of partial fractal mapping and introduce a general framework for a class of fractal-based hybrid image coding systems. In addition, a real hybrid system is presented where non-fractal-coded regions are used to help the encoding of fractal-coded regions. Experiments show that our hybrid system outperforms the JPEG image compression standard at high compression ratios. We believe that under our general framework, the compression result can be further improved by combining fractal technique with other more sophisticated methods.

Fig. 5. Original and decoded 'Lena' images. Upper-left: Original image; Upper-right: Quadtree decoded image, Compression ratio = 35.6, PSNR = 28.6 dB; Lower-left: JPEG decoded image, Compression ratio = 36.6, PSNR = 29.5 dB; Lower-right: Decoded image by the hybrid system, Compression ratio = 37.9, PSNR = 30.4 dB.

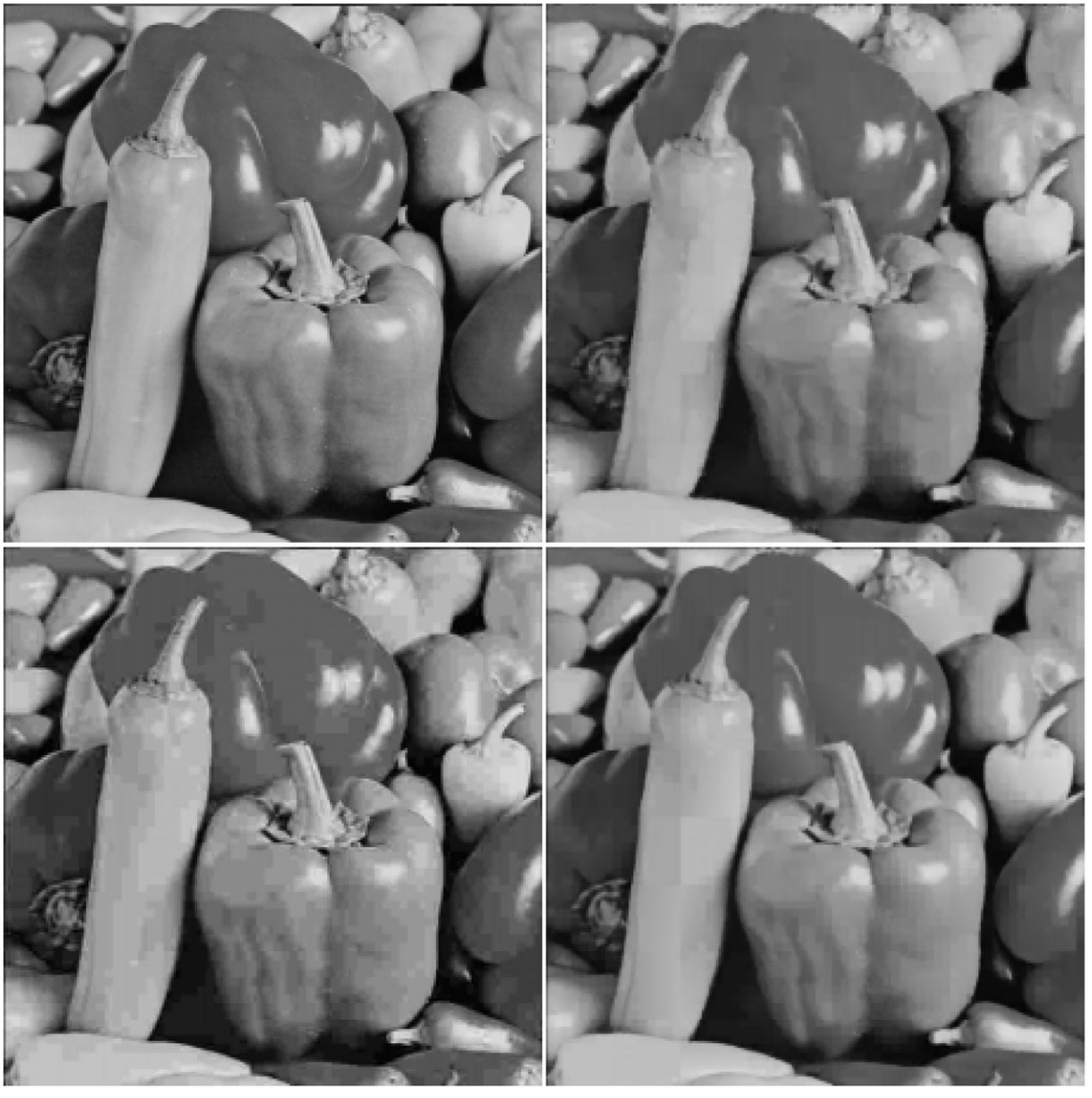

Fig. 6. Original and decoded 'Peppers' images. Upper-left: Original image; Upper-right: Quadtree decoded image, Compression ratio = 34.3, PSNR = 28.7 dB; Lower-left: JPEG decoded image, Compression ratio = 34.0, PSNR = 29.7 dB; Lower-right: Decoded image by the hybrid system, Compression ratio = 35.0, PSNR = 30.5 dB.

Fig. 7. Enlarged areas of 'Lena' image. Left: decoded image areas by the hybrid system; Right: quadtree decoded image areas.

References

[1] M.F. Barnsley, Fractals Everywhere, Academic Press, New York, 1988.

[2] Y. Fisher, Fractal image compression with quadtrees, in: Y. Fisher (Ed.), Fractal Image Compression: Theory and Applications to Digital Images, Springer, New York, 1994.

[3] Y. Fisher (Ed.), Fractal Image Compression: Theory and Application, Springer, New York, 1994.

[4] A.E. Jacquin, Image coding based on a fractal theory of iterated contraceptive image transformations, IEEE Trans. Image Process. 1 (1) (January 1992) 18–30.

[5] A.E. Jacquin, Fractal image coding: a review, Proc. IEEE 81 (10) (October 1993) 1451–1465.

[6] T. Laurencot, A.E. Jacquin, Hybrid image block coders incorporating fractal coding and vector quantization, with a robust classification scheme, AT&T Tech. Memo., February 1992.

[7] Y. Linde, A. Buzo, R.M. Gray, An algorithm for vector quantizer design, IEEE Trans. Commun. COM-28 (January 1980) 84–95.

[8] B.B. Mandelbrot, The Fractal Geometry of Nature, Freeman, San Francisco, 1983.

[9] G.E. Øien, S. Lepsøy, Fractal image coding with fast decoder convergence, Signal Processing 40 (1994) 105–117.

[10] G. K. Wallace, The JPEG still picture compression standard, Commun. ACM 34 (4) (April 1991) 30–44.

[11] Z. Wang, Y.L. Yu, Fractal block coding in residue domain, China. J. Electron. 14 (3) (1997) 236–240.

[12] I.H. Witten, R.M. Neal, J.G. Cleary, Arithmetic coding for data compression, Commun. ACM 30 (6) (June 1987) 520–540.

[13] X.K. Zhou (Ed.), Practical Microcomputer Image Processing, Beijing University of Aeronautics and Astronautics Press, Beijing, 1994.