Fractal Image Coding: A Review

ARNAUD E. JACQUIN, MEMBER, IEEE

*Specific classes of **Fractals** can be used for **Digital Image Compression** or **Coding**. For such an application, the general problem statement is the following. Given any original discrete image specified by an array of pixels, how can a computer construct a fractal image—the coded image—which is both visually close to the original one, and has a digital representation which requires fewer bits than the original image.*

*In this paper, we describe an approach to image coding based on a fractal theory of iterated contractive transformations defined piecewise. The main characteristics of this approach are that: i) it relies on the assumption that image redundancy can be efficiently captured and exploited through piecewise **self-transformability** on a block-wise basis, and ii) it approximates an original image by a **fractal image**, obtained from a finite number of iterations of an image transformation called a **fractal code**. We refer to this approach as **Fractal Block Coding**.*

*The general coding–decoding system is based on the construction, for an original image to encode, of a **fractal code**—a contractive image transformation for which the original image is an approximate fixed point—which, when applied iteratively on any initial image at the decoder, produces a sequence of images which converges to a fractal approximation of the original. The fractal code consists of a description of both an image partition and a contractive image transformation defined as a list of parent and child block transformations, each specified by a small set of quantized parameters. We describe the design of such a system for the encoding of monochrome digital images at rates below 1 b/pixel, without any entropy coding of the parameters of the fractal code. We also present novel ideas and extensions from the work of a number of researchers which has appeared since the publication of the fractal block coding work of the author.*

I. INTRODUCTION

Deterministic *Fractals* have the intrinsic property of having extremely high visual complexity while being very low in information content, as they can be described and generated by simple recursive deterministic algorithms [38]. They are mathematical objects with a high degree of redundancy in the sense that they are recursively made of transformed copies of either themselves or parts of themselves. These objects, which arise from the mathematical theory of *Iterated Sequences*, were first labeled mathematical "curiosities" or "monsters" by mathematicians in the beginning of the twetieth century who lacked the tools to properly analyze and understand them [12], [14], [15]. After falling into nearly complete oblivion for a while, they were "rediscovered" by the mathematical research community in the 1970's, thanks to the pioneering work of Mandelbrot who also coined their name [38]. It is indisputable that this rediscovery was also triggered by the availability of computers and automatic graphic tools which made it possible for the first time to render and visualize them as complex, beautiful, often realistic-looking objects or scenes [38], [47]. For the past ten years, fractals have also been part of a set of tools in a variety of fields in Physics, where they are closely related to *Chaos Theory* [19], [49]. They have recently emerged in various fields of Electrical Engineering, as attested by the contents of this Special Section.

Fractal-based techniques have been applied in several areas of digital image processing, such as image segmentation [50], image analysis [51], [58], image synthesis and computer graphics [4], [7], [17], [22], [23], [48], [54], and texture coding [37], [55]. Barnsley was the first to propose the notion of *Fractal Image Compression*, by which real-life objects or images would be modeled by deterministic fractal objects—attractors of sets of two-dimensional affine transformations [2]–[4]. The mathematical theories of *Iterated Function Systems (IFS)* and *Recurrent Iterated Function Systems* [5], [6], along with the important *Collage Theorem*, constitute the broad foundations of fractal image compression. However, these theories alone do not provide any constructive procedure for the "encoding" of a gray-tone image—as understood by the image coding community—i.e., in an automated way. This particular task can be performed by defining piecewise affine contractive transformations which make use of only the *partial self-transformability* of images.

In this paper, we are only concerned with fully automated, robust, block-based fractal image coding schemes which can *compress* or *encode* any digital monochrome image, and can therefore truly be called image compression schemes [20], [28], [42], [52]. The principles of *Fractal Block Coding* described here were originally published in

Manuscript received July 8, 1993. Parts of this paper are directly excerpted from a paper previously published by the author in the IEEE TRANSACTIONS ON IMAGE PROCESSING (vol. 1, no. 1, Jan. 1992), entitled "Image coding based on a fractal theory of iterated contractive image transformations." The work described in this paper was performed while the author was in the Mathematics Department, Georgia Institute of Technology, Atlanta, GA.

The author is with AT&T Bell Laboratories, Murray Hill, NJ 07974-0636.

IEEE Log Number 9212425.

[25]–[27]. They are rooted in a general theory of iterated contractive transformations in metric spaces of images based on the work of Barnsley *et al.* [2]–[6]. The purpose of this paper is twofold. Sections II–V contain the mathematical principles as well as the description of a specific fractal image coding scheme; they are meant to acquaint readers with block-based fractal image coding. Section VI is a compilation of novel ideas and extensions which have been introduced by researchers following the publication of the initial work of the author.

The basic mathematical theory is succinctly presented in Section II. Section III addresses practical issues in the design of a digital image coder based on iterated transformation theory. This covers the construction of a partition of an image support, the selection of a distortion measure, and the specification of a class of discrete contractive image transformations defined blockwise from which fractal codes are selected. In Section IV, we describe the procedure for the encoding of any monochrome digital image, given a specific fractal block coder. The procedure consists of the organized search of a "virtual codebook" obtained from a pool of domain blocks and a pool of block transformations. The result of this search is a discrete image transformation defined blockwise, which is built so as to leave the original image approximately invariant and which we call a *Fractal Code*. In Section V, we address the decoding or reconstruction of an image from a fractal code, present and analyze coding simulations, and show how to evaluate bit rates. Results are given for the coding of 512 × 512 8-b/pixel digital images such as "Lena." We also point to the analogies which exist with an image coding technique known as *Vector Quantization* (VQ), and indicate the specificity of fractal block coding. In Section VI, we present novel ideas and extensions from the work of a number of researchers [8]–[10], [16], [24], [30]–[33], [35], [36], [39], [41], [43]–[46], [57], which have appeared since the publication of the original fractal image coding work by the author.

II. Theoretical Foundations

In this section, we briefly present the underlying mathematical principles of fractal image coding based on a theory of contractive iterated image transformations. For a broader introduction to data compression using fractal techniques, including the one-dimensional case of contour encoding, we refer the reader to a paper by Beaumont [8].

A. The Inverse Problem of Iterated Transformation Theory

Let (M, d) denote a metric space of digital images, where d is a given metric—distortion measure, and let μ_{orig} be an original image that we want to encode. The *inverse problem* of iterated transformation theory is the construction of a contractive image transformation τ, defined from the space (M, d) to itself, for which μ_{orig} is an approximate *fixed point*. We denote by $\mathcal{F}$ the set of allowed transformations: a specific subset defined *a priori* of the space of all contractive transformations in (M, d). The requirements on the transformation τ are formulated as follows:

$$\exists s < 1 \text{ such that } \forall \mu, \nu \in M \quad d(\tau(\mu), \tau(\nu)) \leq s d(\mu, \nu) \tag{1}$$

and

$$d(\mu_{\text{orig}}, \tau(\mu_{\text{orig}})) \text{ is as "close to zero" as possible.} \tag{2}$$

The scalar s is called the *contractivity* of τ. Under these conditions, and provided that τ has a lower complexity than the original image, τ can be seen as a *lossy*[1] *image code* for μ_{orig}. By repeated application of the triangular inequality [13] in (M, d) and use of the contractivity of τ, it is easy to show that, for any image μ_0 and any positive integer n:

$$d(\mu_{\text{orig}}, \tau^n(\mu_0)) \leq \frac{1}{1-s} d(\mu_{\text{orig}}, \tau(\mu_{\text{orig}})) + s^n d(\mu_{\text{orig}}, \mu_0). \tag{3}$$

From (2) and (3), and since $s < 1$, we see that after a number of initial iterations the terms of any iterated sequence of the form:

$$\{\mu_n = \tau^n(\mu_0)\}_{n \geq 0} \tag{4}$$

where μ_0 is some arbitrary initial image, cluster around the original image. In a space of quantized discrete images, the sequence converges exactly to a stable image, which as a result of its iterative construction, is said to be *fractal* [4], [38].

Notes:

1) The closeness of $\tau^n(\mu_0)$ to μ_{orig} is conditioned by the distortion $d(\mu_{\text{orig}}, \tau(\mu_{\text{orig}}))$.

2) From (3), it is clear that the convergence of the sequence depends fundamentally on s being strictly smaller than one.

3) The constant transformation $\tau = \mu_{\text{orig}}$, which has a contractivity zero and which clearly satisfies (1) and (2), will never be in $\mathcal{F}$ as we are only interested in transformations which can be unambiguously described with fewer bits than the original image—a requirement for compression.

We call such a transformation τ in $\mathcal{F}$ which satisfies (1) and (2) a *fractal code* for μ_{orig}, and say that μ_{orig} is approximately *self-transformable* under τ. This terminology is used because images produced by the above procedure are the result of the iterated application of a deterministic image transformation to an initial image, a procedure which is typical of the construction of deterministic fractal objects [4], [38]. A block diagram of a very general fractal coding–decoding system based on these principles is given in Fig. 1. We now describe the structure of the image transformations that can be used as fractal codes.

[1] As opposed to "lossless"; from τ, only a visually close approximation to μ_{orig} can be recovered.

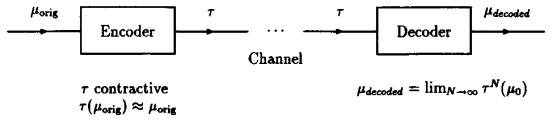


Fig. 1. A general fractal encoding–decoding system.

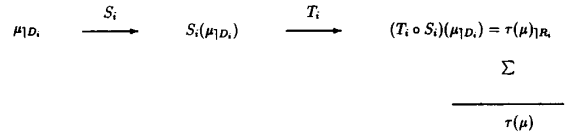


Fig. 2. Application of an image transformation τ defined blockwise to an image μ.

B. Structure and Construction of Fractal Codes for Images

The realization that the redundancy in digital images can be captured through blockwise self-transformability leads us to focus our attention on a class of image transformations of the general form

$$\forall \mu \in M, \tau(\mu) = \sum_{0\le i<N} (\tau\mu)_{\restriction R_i} = \sum_{0\le i<N} \tau_i(\mu_{\restriction D_i}) \quad (5)$$

where

$$\mathcal{P} = \{R_i\}_{0\le i<N} \quad (6)$$

denotes a nonoverlapping partition of the image support into N *range cells*—usually square blocks of possibly different sizes. The symbol $\mu_{\restriction R_i}$ denotes the restriction of the image μ to the cell R_i; we call it the image block over R_i and write

$$\mu = \sum_{0\le i<N} \mu_{\restriction R_i} \quad (7)$$

simply to indicate that an image is the union of its restrictions to the partition cells. The symbol τ_i denotes an elemental block transformation from the *domain cell* D_i to the *range cell* R_i. For clarity, τ_i is written as the composition of two transformations S_i and T_i

$$\tau_i = T_i \circ S_i \quad (8)$$

where S_i and T_i are the so-called *geometric* and *massic* parts of τ_i, respectively. These will be specified in Section III-C. The application of such an image transformation τ to an image μ is illustrated in the diagram in Fig. 2.

The construction of a fractal code τ for μ_{orig}—a contractive image transformation of the form described above under which μ_{orig} is approximately blockwise self-transformable—will be done by defining the elemental τ_i's separately, and independently from one another. Thus the fractal block encoding of μ_{orig} amounts to finding, for every partition index i, a transformation τ_i from a domain cell D_i to the range cell R_i such that the transformed domain block $\tau_i(\mu_{\text{orig}\restriction D_i})$ is a close approximation of the original range block $\mu_{\text{orig}\restriction R_i}$. We refer to the ith step as the encoding of the ith block of μ_{orig} on the partition $\mathcal{P}$.

III. Design of a Fractal Block Coding System

The three main issues involved in the design of a fractal block coding system are: i) the (image-dependent) partitioning of an image, ii) the choice of a measure of distortion between two images, and iii) the specification of a finite class of contractive image transformations defined consistently with a partition, and of a scheme for the quantization of their parameters. Throughout this section the design choices are those of the author. Other designs and their effects on compression rates and coded image quality will be discussed in Section VI.

A. Image Partitions

The square support S of the original digital image μ_{orig} is partitioned into nonoverlapping square range cells of two different sizes, thus forming a *two-level square partition.* This type of partition is closely related to *quad-trees* [56], [59], [60]. The larger cells—of size $B \times B$—are referred to as (range) *parents*, the smaller ones—of size $B/2 \times B/2$—as (range) *children*. A parent can be split into **up to** four nonoverlapping children. Decisions about the splitting of a parent cell are made during the encoding of the image block over this cell (see Section V-C). A partition constructed in this way is image-dependent; it allows the coder: i) to use larger blocks to take advantage of smoothly varying image areas, and ii) to use smaller blocks to capture detail in complex areas such as rugged boundaries and fine textures.

The selection of sizes and shapes for image cells depends on several factors. Small image blocks—4×4 and below—are i) easy to analyze and to classify geometrically, ii) they allow a fast evaluation of interblock distances, iii) they are easy to encode accurately, and iv) they lead to a robust encoding system—one whose performance is steady, even when source images are diverse. On the other hand, large blocks—5×5 and above—i) allow a better exploitation of the redundancy in smooth image areas, and ii) lead to higher compression ratios.

B. Distortion Measure

We construct a distortion measure between digital images from an interblock distortion measure. Let $S(i_0, j_0, B)$ denote the square block of size $B \times B$, with the bottom left corner at the intersection of image row i_0 and image column j_0. Let μ be an $r \times r$ image, and $\tilde{\mu}$ be an approximation of μ. Let $\mu_{\restriction S}$ and $\tilde{\mu}_{\restriction S}$ denote their restrictions to the cell $S(i_0, j_0, B)$, and $\mu_{i,j}$ the gray level of pixel (i, j). The L_2 or *root-mean-square* distortion between the image blocks $\mu_{\restriction S}$ and $\tilde{\mu}_{\restriction S}$ is defined as the square root of the sum over the cell S, of the squared differences of pixel values, i.e.:

$$d_{L_2}(\mu_{\restriction S}, \tilde{\mu}_{\restriction S}) = \left(\sum_{0\le i,j<B} (\mu_{i_0+i,j_0+j} - \tilde{\mu}_{i_0+i,j_0+j})^2 \right)^{1/2}. \quad (9)$$

We define the distance d between two images as the sum over the partition of all block distortions, and the *peak*

signal-to-noise ratio (SNR) by

$$\mathrm{SNR} = 10 \log_{10} \left(\frac{dr(\mu)^2}{d(\mu, \tilde{\mu})/r^2} \right) \tag{10}$$

where $dr(\mu)$ denotes the dynamic range of μ.

C. A Class of Discrete Image Transformations

In this section, we describe the class $\mathcal{F}$ of discrete contractive affine image transformations defined blockwise that the coder uses.

1) Geometric Part S_i: It consists in the discrete form of the $D : B$ spatial contraction operator S, which maps image blocks from a domain cell $D_i = S(i_d, j_d, D)$, to a range cell $R_i = S(i_r, j_r, B)$. In the simple case where $D = 2B$, the pixel values of the contracted image on the range block $S(i_r, j_r, B)$ are the average values of four pixels in the domain block:

$$\begin{aligned}(S\mu)_{i_r+i,j_r+j} = (&\mu_{I(i),J(j)} + \mu_{I(i)+1,J(j)} \\ &+ \mu_{I(i),J(j)+1} + \mu_{I(i)+1,J(j)+1})/4, \\ &\text{for all } i,j \in \{0, \cdots, B-1\}\end{aligned} \tag{11}$$

where the index functions I and J are defined by:

$$I(i) = i_d + 2i \quad \text{and} \quad J(j) = j_d + 2j. \tag{12}$$

In [25], we show that this operator has contractivity one.

2) Massic Part T_i: Massic transformations are those which process image blocks supported on a range cell R_i. This terminology is used in order to stress the fact that these transformations affect the pixel values of the block. A few simple transformations defined in the space of discrete image blocks supported on a $B \times B$ range cell are listed below:

i) Absorption at gray level $g_0 \in \{v_k\}_{0 \le k < G}$:

$$(\theta\mu)_{i,j} = g_0. \tag{13}$$

ii) Luminance shift by constant $\Delta g \in \{\pm v_k\}_{0 \le k < G}$:

$$(\tau\mu)_{i,j} = \mu_{i,j} + \Delta g. \tag{14}$$

iii) Contrast scaling by $\alpha \in [0, 1]$:

$$(\sigma\mu)_{i,j} = \alpha\mu_{i,j}. \tag{15}$$

The following transformations simply *shuffle* pixels within a range block, in a deterministic way—we call them *isometries*. We list in the following a list of the eight canonical isometries of a square block.

1) Identity:

$$(\iota_0\mu)_{i,j} = \mu_{i,j}. \tag{16}$$

2) Orthogonal reflection about mid-vertical axis ($j = (B-1)/2$) of block:

$$(\iota_1\mu)_{i,j} = \mu_{i,B-1-j}. \tag{17}$$

3) Orthogonal reflection about mid-horizontal axis ($i = (B-1)/2$) of block:

$$(\iota_2\mu)_{i,j} = \mu_{B-1-i,j}. \tag{18}$$

Table 1 L_2 Contractivities of the Massic Transformations of Section III-C

Massic Transformation	L_2 Contractivity
Absorption at g_0	$s = 0$
Luminance shift by Δg	$s = 1$
Contrast scaling by α	$s = \alpha^2$
Color reversal	$s = 1$
Isometries $\{\iota_k\}_{0 \le k \le 7}$	$s = 1$

4) Orthogonal reflection about first diagonal ($i = j$) of block:

$$(\iota_3\mu)_{i,j} = \mu_{j,i}. \tag{19}$$

5) Orthogonal reflection about second diagonal ($i + j = B - 1$) of block:

$$(\iota_4\mu)_{i,j} = \mu_{B-1-j,B-1-i}. \tag{20}$$

6) Rotation around center of block, through $+90°$:

$$(\iota_5\mu)_{i,j} = \mu_{j,B-1-i}. \tag{21}$$

7) Rotation around center of block, through $+180°$:

$$(\iota_6\mu)_{i,j} = \mu_{B-1-i,B-1-j}. \tag{22}$$

8) Rotation around center of block, through $-90°$:

$$(\iota_7\mu)_{i,j} = \mu_{B-1-i,j}. \tag{23}$$

Some other, more complex, shuffling transformations can be constructed. In effect, massic transformations allow to generate from a single block a whole *family* of geometrically related transformed blocks, which provides a pool in which matching blocks will be looked for during the encoding.

The L_2 contractivities of the discrete block transformations described in this section can be easily computed; the results are given in Table 1. As an example, we derive below the contractivity of a contrast scaling by α. As in Section III-B, S denotes the cell $S(i_0, j_0, B)$:

$$\begin{aligned} &d(\sigma(\mu_{\restriction S}), \sigma(\nu_{\restriction S})) \\ &\quad = \left(\sum_{0 \le i,j < B} (\alpha\mu_{i_0+i,j_0+j} - \alpha\nu_{i_0+i,j_0+j})^2 \right)^{1/2} \\ &\quad = \alpha d(\mu_{\restriction S}, \nu_{\restriction S}) \end{aligned} \tag{24}$$

so the contractivity of this particular block transformation is α.

IV. Construction of Fractal Codes for Digital Images

This section addresses the implementation of a specific fractal block coder based on the theoretical coding procedure described in Section II, and on the design choices made in Section III. Here, we describe the step-by-step construction of a fractal block code for any given original image.

A. Overview of the Procedure

An original $r \times r$ digital image μ, quantized to 256 gray levels, is given as input to the coder. Let $\{R_i\}_{0 \le i < N}$ denote the image partition made of range cells of two different sizes.[2] Recall that an image transformation τ has the form:

$$\tau = \sum_{0 \le i < N} \tau_i, \quad \text{with } \tau_i = T_i \circ S_i.$$

Given a range cell R_i (of size $B \times B$), the construction of a transformation τ_i which maps *onto* this cell is broken into two distinct steps corresponding to the transformations S_i and T_i, respectively.

- The construction of the spatial contraction S_i amounts to selecting an image domain block $\mu_{\restriction D_i}$ of size $D \times D$, which will be contracted to a block $S_i(\mu_{\restriction D_i})$ of size $B \times B$. It is important to note that the specification of the domain cell D_i (location and size) is equivalent to the description of the spatial contraction S_i (translation vector and contraction factor).
- The second part of the construction of τ_i consists of selecting the proper *processing* of the contracted domain block $S_i(\mu_{\restriction D_i})$, i.e., of finding the block transformation T_i which minimizes the distortion between $T_i \circ S_i(\mu_{\restriction D_i})$ and $\mu_{\restriction R_i}$.

We can thus formally define a pool of domain blocks D, made up of all image blocks which can be extracted from the original image, and which are larger than the range block R_i. We can also formally define a pool of massic transformations $\mathcal{T}$, made of all discrete block transformations T_i. The encoding of the range block $\mu_{\restriction R_i}$ consists in finding a "best pair" $(D_i, T_i) \in \mathcal{D} \times \mathcal{T}$, which is such that the distortion

$$d(\mu_{\restriction R_i}, T_i \circ S_i(\mu_{\restriction D_i})) \text{ is minimum.} \tag{25}$$

We call the product pool $\mathcal{D} \times \mathcal{T}$ a *global pool* or *virtual codebook*[3]. The encoding of a range block is illustrated symbolically in Fig. 3.

The following two sections address a way to conduct a *directed search* for a best pair (D_i, T_i) by *a priori discarding* large subsets of the global pool, and by *analyzing* the range block to encode along with a set of candidate domain blocks.

B. Pools of Domain Blocks

The maximal domain pool corresponding to a range block of size $B \times B$ can be thought of as all image blocks of size $D \times D$ $(D > B)$ located anywhere in the image to encode. It is typically very large, but it can be trimmed and organized in order to make the search for an optimal domain block tractable.

An initial domain pool $\mathcal{D}$ can be obtained by sliding a window of size $D \times D$ $(D = 2B$ is used) across the original image. The window is first located with its bottom left corner at $(0, 0)$. It then moves from one position to the next by steps of either δ_h pixels horizontally to the right, or δ_v pixels vertically upwards, in such a way that it remains entirely inside the image support, at all times. The steps can typically be chosen equal to B or $B/2$.

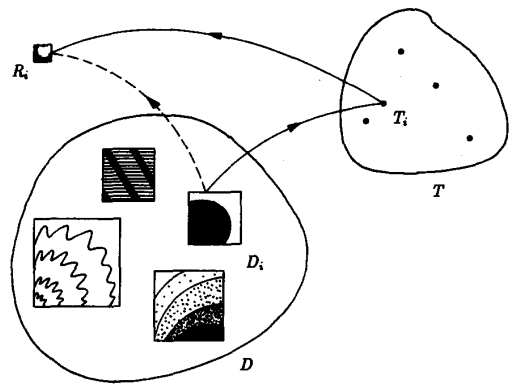


Fig. 3. Encoding of the range block R_i.

The domain blocks in this pool $\mathcal{D}$ are then classified, based on their perceptual geometric features. Using the thorough study of image block classification done by Ramamurthi and Gersho [53], we define three types of blocks, and create three classes $\mathcal{D}_s, \mathcal{D}_e$, and $\mathcal{D}_m$, for *shade blocks, edge blocks*, and *midrange blocks*, respectively. A shade block is "smooth" with no significant gradient. An edge block presents a strong change of intensity across a curve—often a piece of object boundary, which runs through the block. A midrange block has moderate gradient but no definite edge. It is assumed to be approximately isotropic, i.e., not to have any privileged orientation—"texture blocks" belong to this category. The total domain pool $\mathcal{D}$ is the union

$$\mathcal{D} = \mathcal{D}_s \cup \mathcal{D}_e \cup \mathcal{D}_m \tag{26}$$

where $\mathcal{D}_e$ is further split into two subclasses of *simple* edge blocks $\mathcal{D}_{se}$, and *mixed edge* blocks $\mathcal{D}_{me}$:

$$\mathcal{D}_e = \mathcal{D}_{se} \cup \mathcal{D}_{me}. \tag{27}$$

Details about the classification algorithm can be found in [25], [53].

Shade blocks are not used as domain blocks, since i) a shade block remains a shade block under any of the block transformations that we consider, and ii) absorptions generate uniform shade blocks. We can therefore remove them from the pool. Moreover, it is easy to check that the nature of a domain block is preserved under application of any element of the transformation pool.

C. Pools of Transformation

We now propose a procedure for the construction of a transformation pool $\mathcal{T}$, and for the directed search of the resulting global pool $\mathcal{D} \times \mathcal{T}$. The quantization of the parameters of the transformations is dictated by storage requirements. The description of a block transformation should be kept as simple as possible in order to obtain low

[2] The actual construction of this two-level partition is addressed in Section V-C.

[3] The analogy with the notion of codebook used in vector quantization for image coding is discussed in Section V-D.

bit rates. Note that the contractivities of the transformations can be obtained by multiplying together the contractivities of the elemental transformations which compose them.

For every range block $\mu_{\restriction R_i}$ in the original image, our encoding procedure consists, according to the nature of the block, of one of the following.

Case 1: $\mu_{\restriction R_i}$ is a shade block.

We simply approximate $\mu_{\restriction R_i}$ by a uniformly gray block, with gray level equal to the average gray level of $\mu_{\restriction R_i}$, which is denoted $\bar{\mu}_{\restriction R_i}$. The transformation τ_i which generates this uniform block is simply the absorption at $g_i = \text{quant}\,(\bar{\mu}_{\restriction R_i})$, where the function quant represents uniform quantization with 6 b in the range $[0, 255]$.

Case 2: $\mu_{\restriction R_i}$ is a midrange or an edge block.

Every element $\mu_{\restriction D_j}$ of the domain pool of the same type as the block $\mu_{\restriction R_i}$ is scanned. The analysis of a pair $(\mu_{\restriction R_i}, \mu_{\restriction D_j})$ determines the selection of the transformation τ_i used to encode $\mu_{\restriction R_i}$.

Case 2a): $\mu_{\restriction R_i}$ is a midrange.

For midrange blocks, we restrict our attention to massic transformations that are compositions of a contrast scaling and a luminance shift, of the form

$$T_i(S_j\mu_{\restriction D_i}) = \alpha_i(S_j\mu_{\restriction D_i}) + \Delta g_i \tag{28}$$

where α_i is a contrast scaling factor which is allowed to take values in a set of four values smaller than one, and Δg_i is computed so that the average gray levels of the range block and the scaled domain block are approximately the same, i.e.,

$$\Delta g_i = \text{quant}\,(\bar{\mu}_{\restriction R_i} - \alpha_i\bar{\mu}_{\restriction D_j}) \tag{29}$$

where this time, the function quant represents uniform quantization with 6 b in the range $[-128, 127]$.

Case 2b): $\mu_{\restriction R_i}$ is an edge.

The analysis of the pair $(\mu_{\restriction R_i}, \mu_{\restriction D_j})$ consists of a crude segmentation of the blocks $\mu_{\restriction R_i}$ and $S_j\mu_{\restriction D_j}$, based on the computation of histograms of gray levels [20], [25]. We assume that image blocks of sufficiently small dimension can be segmented into two uniform regions, one *dark*, one *bright*, separated by a transition region. The segmented blocks are denoted $\mu^{\text{seg}}_{\restriction R_i}$, and $S_j\mu^{\text{seg}}_{\restriction D_j}$, respectively. After this segmentation is performed, we compute the dynamic range of the segmented blocks as the gray level difference between the bright and dark regions.

We use massic transformations that are compositions of a contrast scaling, a luminance shift, and an isometry, of the form:

$$T_i(S_j\mu_{\restriction D_j}) = \iota_{n_i}(\alpha_i(S_j\mu_{\restriction D_j}) + \Delta g_i). \tag{30}$$

First, α_i is computed so that $\mu^{\text{seg}}_{\restriction R_i}$ and $S_j\mu^{\text{seg}}_{\restriction D_j}$ have the same dynamic range

$$\alpha_i = \frac{dr(\mu^{\text{seg}}_{\restriction R_i})}{dr(S_j\mu^{\text{seg}}_{\restriction D_j})}. \tag{31}$$

It is then quantized to the nearest value in a set of four values smaller than one. Secondly, Δg_i is computed and quantized so that the gray levels of either the dark regions, or the bright regions—depending on the dominant populations of $\mu^{\text{seg}}_{\restriction R_i}$ and $S_i\mu^{\text{seg}}_{\restriction D_j}$—are the same. Finally, the one among the eight isometries $\{\iota_{n_i}\}_{0\le n_i\le 7}$ which minimizes the distortion measure is selected.

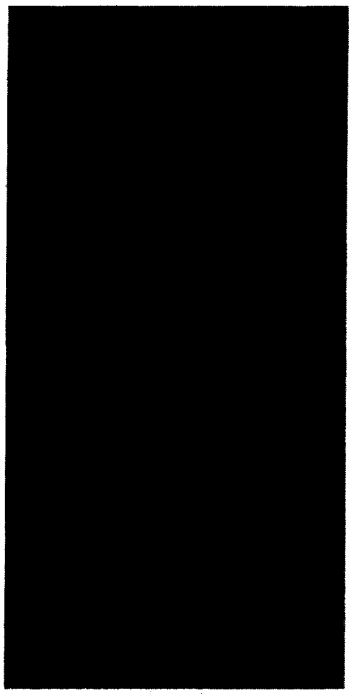


Fig. 4. First seven iterations of the "peppers-to-Lena" decoding sequence.

The analysis of the pair $(\mu_{\restriction R_i}, S_j\mu_{\restriction D_j})$, and the related construction of a candidate massic transformation T_i are done for every domain $\mu_{\restriction D_j}$ in the domain pool. The domain block $\mu_{\restriction D_i}$ and associated massic transformation T_i which minimize the distortion $d(\mu_{\restriction R_i}, T_i \circ S_i(\mu_{\restriction D_j}))$ are finally selected. The optimal transformed domain block $\tau_i(\mu_{\restriction D_i})$ is called the *matching block* for $\mu_{\restriction R_i}$.

V. Results

A. Image Reconstruction from a Fractal Code

We use the original 512 × 512, 8-bpp image "Lena" (Fig. 8), and a fractal code obtained through the encoding procedure described in Section IV. The natural decoding scheme simply consists in iterating the fractal code τ on any initial image μ_0, until convergence to a stable decoded image is observed.

The sequence of images $\{\mu_n = \tau^n(\mu_0)\}_{n=0}^{\infty}$ is called the *reconstruction sequence* for the code τ, with initial image μ_0. The mapping of an image under the fractal code is done sequentially. For each cell index i, the transformation τ_i is applied to the current image block over the domain cell D_i, and mapped onto the range cell R_i. In Fig. 4 we show the first eight iterations of a decoding sequence corresponding to a fractal block code for the 512 × 512 "Lena" image, with the initial image "peppers."

It is interesting to consider another decoding scheme which illustrates the convergence of the iterative reconstruction procedure differently. By looking at the form of the transformations in $\mathcal{F}$, we can see that they are *affine* [13] in the normed vector space $(N, \|\cdot\|)$ associated with the metric space (M, d); i.e., a code τ has the form:

$$\forall \mu \in N, \tau(\mu) = L(\mu) + \nu_0 \tag{32}$$

where L is a contractive *linear* image transformation, and ν_0 is an image which is blockwise uniform. The Nth iterate of the reconstruction sequence can be written as

$$\mu_N = \sum_{i=0}^{N-1} L^i(\nu_0) + L^N(\mu_0). \tag{33}$$

The above expression for μ_N is a *series expansion* of order $N-1$. The term $L^i(\nu_0)$ is the ith order term of the expansion. The finite sum

$$\sum_{i=0}^{N-1} L^i(\nu_0) \tag{34}$$

is the *partial sum* of order $N-1$, and the residual term

$$L^N(\mu_0) \tag{35}$$

is the *remainder* of the expansion.

By a result of Banach space theory [13], it can be easily shown that

$$\|L^N(\mu_0)\| \leq s^N \|\mu_0\| \tag{36}$$

which, since $s < 1$, is negligible for sufficiently large values of N. Note that the remainder is the only term in the expression of μ_N which depends on the initial image μ_0. Because of the L_2-contractivity of τ, the L_2-energy of these terms decreases to zero; i.e., the images fade to black. When the running sums of increasing order

$$\nu_0, \sum_{i=0}^{1} L^i(\nu_0), \sum_{i=0}^{2} L^i(\nu_0), \cdots, \sum_{i=0}^{7} L^i(\nu_0) \tag{37}$$

of these terms are computed, the reconstruction sequence corresponding to a uniformly black initial image is obtained. The first eight terms of this running sum—for the fractal code as in Fig. 4—are displayed in Fig. 5.

Values of the SNR between the original "Lena" image and successive terms of this last reconstruction sequence are listed in Table 2. Convergence is obtained within 0.1 dB of accuracy at the twelfth iteration. It turns out that this required number of iterations remains approximately the same for different initial images.

B. Test Results

Information about the specific fractal block coding system used for the encoding of "Lena"—design specifications, encoding procedure, and system performance—is given in Table 3. Under the heading "Encoding specifications," we define: i) the partition of the image support, ii) the pool of domain blocks, and iii) the pool of transformations. Under the heading "System performance" we list: i) the bit rate[4], and ii) the SNR's between the original and the decoded image.

Fig. 5. First eight iterations of the "black-to-Lena" decoding sequence (initial all-black image not shown).

Table 2 Peak SNR Values Between the Original "Lena" and Successive Terms of a Reconstruction Sequence with an Initial Black Image.

It. No.	1	2	3	4	5	6	7	8	9	10
SNR	10.7	14.1	17.4	20.8	24.0	26.8	28.9	30.2	30.8	31.1

A coding simulation was carried out with an empirically chosen, fixed set of coding specifications. The original image is the standard 512 × 512, 8-bpp "Lena" image shown in Fig. 8. It was initially split into four 256 × 256 subimages, which were encoded independently from one another. We denote by τ_{Lena} the optimal fractal block code obtained for "Lena." The image $\tau_{\text{Lena}}(\mu_{\text{orig}})$, present at the encoder, and which indicates the image quality that can expected at the decoder (see (3)), is shown in Fig. 9.

The decoded image, a fractal approximation of the original, is displayed in Fig. 10.

The enhanced error image, equal to five times the pixel-to-pixel difference between the original and the decoded image, is displayed in Fig. 11.

The performance of the encoding system, in terms of bit rates and fidelity of the decoded image to the original, is

[4]Section V-C addresses the way bit rates are computed, as the ratio of the number of bits necessary to describe the fractal code divided by the number of pixels in the image.

Table 3 Design Specifications and Performance of a Fractal Block Coding System

Encoding Specifications	
Partition	
Subimage size:	256
Type of partition:	two–level hierarchical square
Range blocks:	8×8 (parent),4×4 (child)
Domain Pool	
Domain blocks:	16×16 (parent) 8×8 (child)
Domain block shifts:	$\delta_h = 4, \delta_v = 4$
Classification:	shade (s), midrange (m), simple edge (se),
	mixed edge (me)
Transformation Pool	
Shade block:	Absorptions
Midrange block:	Contrast scalings
	Luminance shifts
Edge block:	Contrast scalings
	Luminance shifts
	Isometries
System Performance	
Test image	"Lena", 512×512, 8 bpp
Bit rate	0.06 bpp
SNR (peak)	31.4 dB
SNR (signal energy to noise)	24.2 dB

given in Table 3. Histograms of the parameters of the parent and child transformations for τ_{Lena} are plotted in Figs. 6 and 7.

The following features of these histograms are of interest[5]: i) the histograms of domain-to-range distances peak at zero, indicating that optimal domain blocks are often found in the immediate vicinity of the range blocks, ii) the histograms of luminance shifts are approximately Laplacian, centered near zero, iii) the seven isometries of a square block listed in Section III-C as ι_1 through ι_7—i.e., the nontrivial rotations and reflections—are rather seldom used.

The decoded image has an overall very good fidelity to the original. Image "textures" are well preserved although some finely textured areas, such as the turban around the hat, are slightly "smoothed out" by the encoding process. Sharp, contrasted contours, whether they are smooth, or rugged, are very accurately preserved. Some artifactual "blockiness" is visible, as is expected for the type of memoryless block coding method used here. It is due to the use of square range blocks. However, the reconstruction is free of edge degradation in the form of a "staircase effect."

It also appears that the quality of fractal codes relies very heavily on block classification, and analysis. Most of the artifacts visible in reconstructed images are rooted either in wrong block classification or in inaccurate block analysis, both of which are most likely to occur either with very large blocks or with blocks with either weak edges or strongly contrasted textures which are difficult to classify.

C. Computation of Bit Rates

The complete description of an image transformation τ, with respect to its storage or transmission requirements, is essential to the evaluation of compression ratios or bit rates. The important factors are: i) the description of the image partition, ii) the nature of the block transformations used, and iii) the quantization of the numerical parameters of these block transformations. No entropy coding of the transformation parameters was performed to obtain further compression.

Three distinct types of block transformations are used, depending on the nature of the range block being coded (shade, midrange, or edge). In order to compute a bit rate given complete specifications of a fractal block coding system, we first compute the information in bits needed to represent a block transformation of each type. These quantities, denoted respectively as I_s, I_m, and I_e, were computed from the specifications of Table 3, and are shown in Table 4. Secondly, we consider the case of a two-level square partition, with parent cells of size $B \times B$, and child cells of size $B/2 \times B/2$. Let N_s, N_m, and N_e denote the total numbers—parents and children—of (range) shade, midrange, and edge blocks in the image.

The coding of a range parent block is performed in the following way. First, a pool of *parent transformations*[6] is searched for a best transformation, according to the procedure presented in Section IV-C. The image of the parent domain block under the best transformation found—the parent matching block—is then split into four children, and distortion measures are computed between these and the child blocks in the original image. Depending on these distortions, the coding of the parent block is or is not pursued to the next level. For every child block such that the distortion is too high, a *child transformation* is constructed and is "appended" to the parent transformation. If either three or four child transformations are needed, the initial parent transformation is discarded altogether, and the block is coded as four child blocks. There are, therefore, twelve possible coding configurations of a parent block, which are represented symbolically in Fig. 12. The configuration can thus be encoded with $I_c = 4$ b. The average bit rate is

$$\frac{N_p I_c + N_s I_s + N_m I_m + N_e I_e}{N_p B^2} \quad \text{bpp} \qquad (38)$$

[5] Some of these features have also been observed and exploited by other researchers in their own implementations of fractal block coders (see Section VI)

[6] When dealing with two-level square image partitions, we distinguish *parent* transformations, which map to image blocks of size $B \times B$ (parent blocks), from *child* transformations, which map to blocks of size $B/2 \times B/2$ (child blocks).

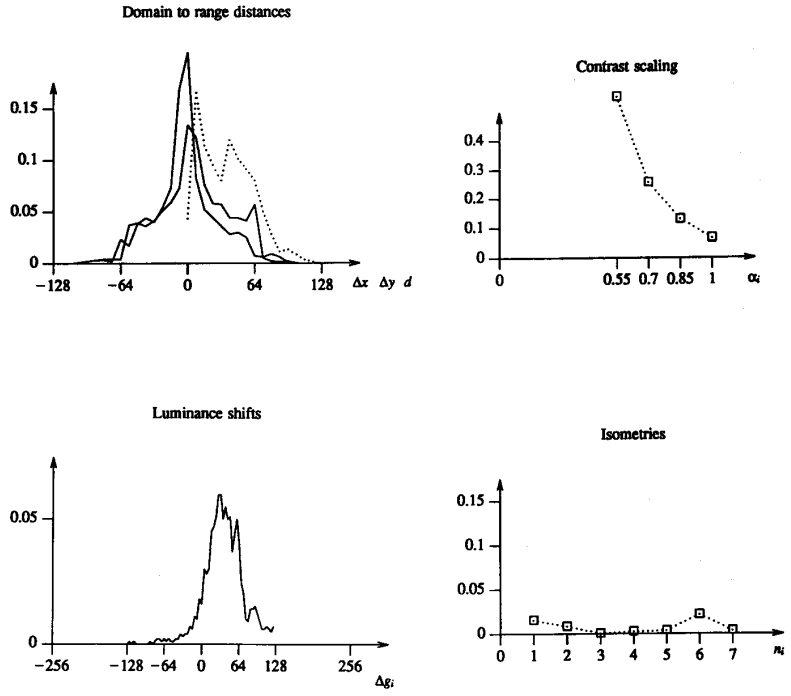


Fig. 6. Statictics of a fractal code (part I: parameters of parent transformations).

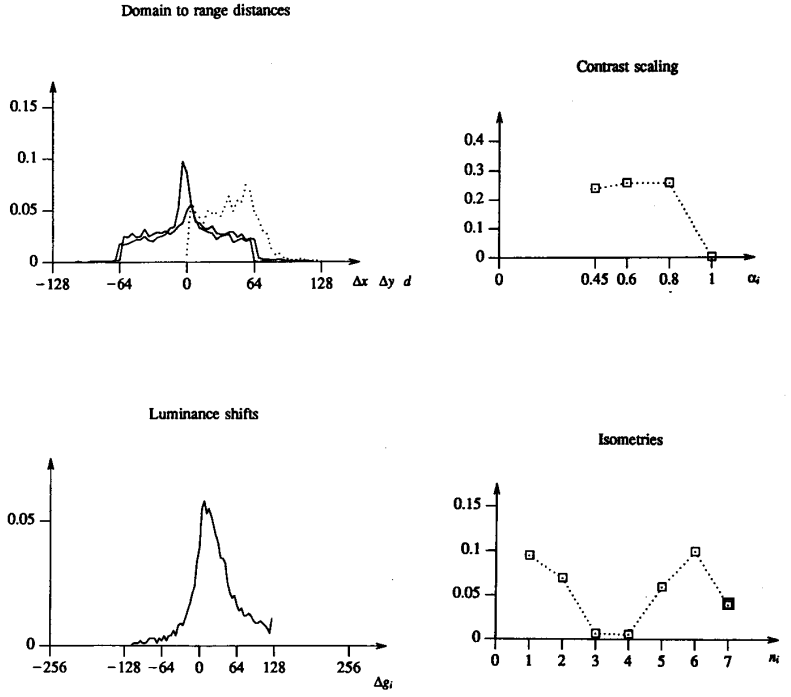


Fig. 7. Statictics of a fractal code (part II: parameters of child transformations).

where $N_p = (r/B)^2$ is the total number of parent blocks in the image.

The fractal code consists of the description of the image partition, along with that of the image transformation defined as the ordered list of range block transformations:

$$\{(\tau_i, \tau_i^{1,1}, \tau_i^{1,2}, \tau_i^{2,1}, \tau_i^{2,2}), \quad 0 \leq i < N\} \qquad (39)$$

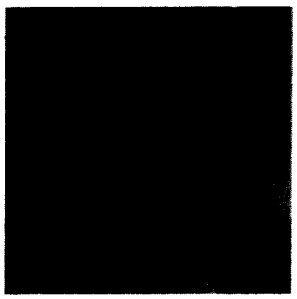


Fig. 8. Original 512 × 512 "Lena" image, 8 bpp.

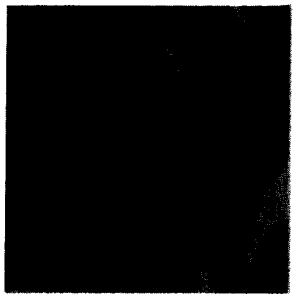


Fig. 9. Encoded "Lena" image $\tau_{\text{Lena}}(\mu_{\text{orig}})$.

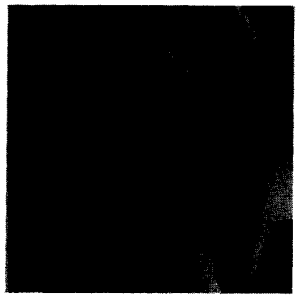


Fig. 10. Decoded "Lena" image obtained after eight iterations of τ_{Lena} on an initial black square, 31.4 dB, 0.6 bpp.

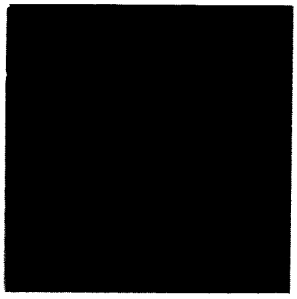


Fig. 11. Enhanced error image for "Lena."

Table 4 Information in Bits for the Representation of Block Transformations

Block Type	Parameters	Information in Bits
Shade	g_i	6
		$I_s = 6$ bits
Midrange	D_i (domain)	parent: $4 + 4 = 8$
		child: $5 + 5 = 10$
	α_i	2
	Δg_i	6
		$I_m = 16$ bits (p)
		$I_m = 18$ bits (c)
Edge	D_i	parent: $4 + 4 = 8$
		child: $5 + 5 = 10$
	α_i	2
	Δg_i	6
	ι_{n_i}	3
		$I_e = 19$ bits (p)
		$I_e = 21$ bits (c)

where $\tau_i^{1,1}, \tau_i^{1,2}, \tau_i^{2,1}, \tau_i^{2,2}$ are four child transformations which are possibly appended to the parent transformation τ_i. Each of these block transformations are from the pools of tranformations defined in Section IV-C, and are specified by a set of quantized parameters. The structure of a fractal code is illustrated in Fig. 13. which the arrows indicate block transformations from domain block to range block.

D. Analogies with Vector Quantization

Fractal block coding is in some sense related to the image block coding technique known as vector quantization (VQ) [1], [18], [21], [53]. In Table 5, we give a comparative list of features shared by both methods. We use VQ terminology, and relate it to the specific terminology introduced in this paper.

The bit rates that we obtain depend partly on the system specifications, and also partly on the complexity of the image to encode. Our coding system is therefore *variable-rate*. However, we found that for given specifications of a coding system, images with comparable block statistics (in terms of percentages of blocks of different types) are encoded with comparable bit rates. Also, the selection of image partitions and allowed classes of block transformations could be made in compliance with a maximum bit rate which would be computed for a "worst encoding case." The particular system described in Sections IV and V is also plagued by high encoding complexity, something which can be remedied by improving the searching strategy for matching blocks (see Section VI).

It is important to stress the following points:

- Domain image blocks are **not present**, and **not needed**, at the decoder; hence the term "virtual codebook." This "codebook" is used only during the encoding phase, for the construction of the fractal code.
- The encoding does not rely on the existence of any

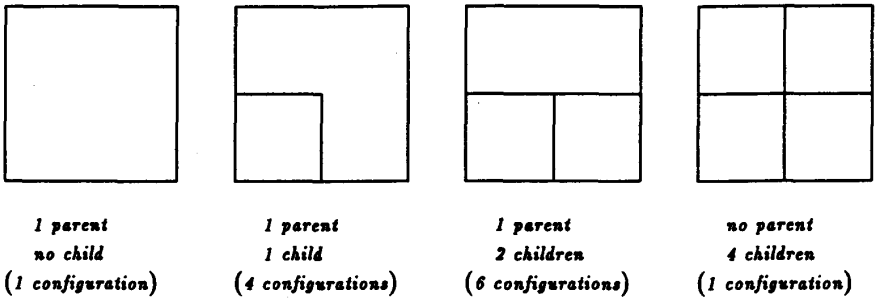


Fig. 12. Coding configurations for a parent range block.

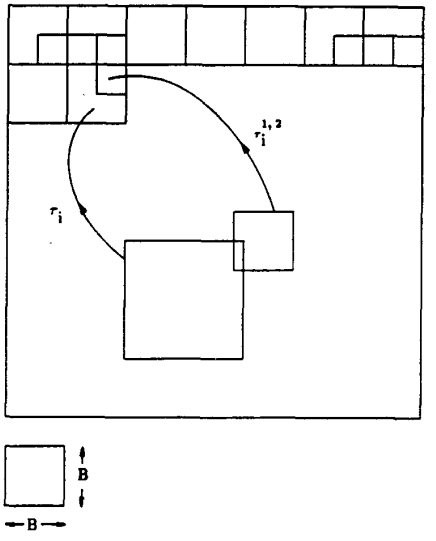


Fig. 13. Structure of a fractal image block code.

"external codebook" but on the redundancy present in the original image itself. The "virtual codebook" is in other words "image-adaptive" in the sense that it is provided by the original image itself.

These features are what make fractal block coding very different from vector quantization. As for the iterative reconstruction of an image, it is characteristic of fractal-based image generation techniques.

Image degradation is of roughly the same type for both types of block-coding methods, although sharp edges are very well preserved by our system. It is interesting to note that the largest block size about which strong image degradation begins to occur is the same in both systems. For (child) range blocks of size 5×5, the crudeness of the block classification, analysis, and segmentation algorithms used in this paper becomes apparent.

VI. Extensions

In this Section, we present a number of extensions which have been proposed by various researchers [8]–[10], [16], [24], [30]–[33], [36], [39], [41], [43]–[46], [57], since the original fractal image block coding work was published by the author in 1989–1990 [25], [26]. These extensions broadly address the following issues: i) the influence of the type of image partition, pool of block transformations, and optimization of the parameters defining these transformations, ii) the reduction of the complexity of the encoding process—the search for best matching blocks in the image to encode, iii) the comparison and possible merging of fractal block coding with more traditional block-based image coding techniques, and iv) the extension of the theory to the three-dimensional case for fractal block coding of sequences of images. We now present some very recent novel ideas and conclusions from these studies below.

Table 5 Comparative List of Features Shared by VQ and Fractal Block Coding

VQ	Fractal Block Coding
Image Partition	
Square blocks.	Square blocks.
Different block sizes possible.	Different block sizes possible.
Codebook	Global pool (virtual codebook)
Training set of images.	Contracted, processed domain blocks extracted from original image.
Codebook design.	Trimming of domain pool.
Off-line transmission of codebook.	*No transmission of domain blocks.*
Block matching	Encoding of range blocks
Decoding	
Image code: list of addresses of blocks in the codebook.	Fractal code: list of block transformations.
Direct reconstruction by look-up table	Iterative reconstruction.

A. Optimality of Transformation Parameters

From Section IV-C, we recall that the block transformations we used for each block type are affine and of the general form:

$$T \circ S(\mu_{\restriction D}) = \alpha_0 J(S\mu_{\restriction D}) + \Delta g \qquad (40)$$

where the indices corresponding to the particular range and domain block considered have been dropped for simplicity. The operator $(\alpha_0 J \circ S)$ is the *linear part* of the transformation, which corresponds to the *image-adaptive* part of the matching process, whereas the term Δg is the *translation term*. In [25], [26], this latter term was *a priori* restricted to be a flat luminance shift over the range block R.

In [24], Jacobs and Fisher conducted a thorough study to determine the optimal number of bits for the uniform quantization of α_0 and Δg, as well as the best choice for $\alpha_{\max}$, the upper bound on α_0 for all the block transformations which constitute the fractal code. In [16], Fisher *et al.* show that it is not necessary to impose that every block transformation be strictly contractive, i.e., $\alpha_{\max}$ can be chosen to be greater than one, and convergence to a stable decoded image will still hold. The study in [24] leads to the optimal choice of uniform quantization of α_0 and Δg with 5 and 7 b, respectively, along with a value of $\alpha_{\max}$ equal to 1.5.

B. Other Block Transformations and Search Complexity Reduction

In [43], [44], Øien, Lepsøy, and Ramstad suggested that the translation term Δg be chosen in a three-dimensional subspace of $\mathcal{R}^{B\times B}$, i.e., be of the form

$$\Delta g = \sum_{k=1}^{k=3} \alpha_k A_k \tag{41}$$

where $\{\alpha_1, \alpha_2, \alpha_3\}$ are real coefficients, and $\{A_1, A_2, A_3\}$ are the fixed basis vectors consisting of the three matrices

$$A_1(i,j) = 1 \quad A_2(i,j) = i \quad A_3(i,j) = j, \quad \text{for} \quad 0 \le i,j < B. \tag{42}$$

The constant term Δg can be geometrically visualized as a *tilted plane* in the three-dimensional space formed by the plane of the image with a third dimension for gray levels. Equivalently, this constant term can be viewed as a first-order, two-dimensional polynomial in the row and column indexes, entirely defined by the three coefficients α_1, α_2, and α_3. The authors use an *inner product space* formulation of the problem of optimally deriving the three parameters, which we summarize here.

Candidate matching blocks are written as a linear combination of the processed domain block $J(S\mu_{\uparrow D})$—the image-dependent part of the matching, and the three fixed basis vectors $\{A_1, A_2, A_3\}$. If the basis vectors happened to form an orthogonal set, the optimal coefficients could be easily computed without having to invert matrices. In [44], Øien *et al.* therefore propose to first Gram–Schmidt orthogonalize the fixed-basis vectors $\{A_1, A_2, A_3\}$, then Gram–Schmidt orthogonalize each candidate domain block $J(S\mu_{\uparrow D})$ to the fixed basis. This results in having to recompute only the optimal coefficient α_0 for each new domain block considered, in effect decoupling the optimization of the scalar coefficients for the constant and linear parts in (40) and (41). Furthermore, the authors define *smooth blocks* to be those image blocks which are approximated sufficiently well by an optimal quadratic polynomial in i and j, the row and column indices of pixels in a range block, of the form

$$P(i,j) = a_2 i^2 + a_1 i + b_2 j^2 + b_1 j + c_0 ij + t_0, \quad \text{for} \quad 0 \le i,j < B \tag{43}$$

for which no search is necessary. By allowing optimal quadratic polynomial approximation of smooth blocks, the percentage of image blocks which can be encoded by only a constant part increases dramatically, and the overall search complexity is greatly reduced. In [44], the authors report a reduction in encoding complexity by an order of magnitude.

A subsequent technique proposed by Monro and Dudbridge, and presented in [39]–[41], is motivated by the desire to reduce even more drastically the search for domain blocks—the main factor for long encoding times with block-based fractal image coding. Since these authors allow very few domain blocks to be searched, they, on the other hand, rely on an even "richer" pool of affine block transformations as in (40), but with translation terms of the form

$$\Delta g = \sum_{k=1}^{3} a_k i^k + \sum_{k=1}^{3} b_k j^k + t_0 \tag{44}$$

which are third-order polynomials in i and j, with no cross products. These constant terms, now defined by seven parameters instead of one, are optimally derived by using a least squares criterion. The authors report a significant increase in image quality by allowing these higher order shifts, as compared with the zero-order approximation case.

It was also reported by a number of authors [10], [41] that one could do away with the seven isometries of a square block listed in Section III-C, as ι_1 through ι_7—i.e., all nontrivial rotations and reflections, with no perceivable degradation of image quality but with a great reduction in search complexity. In [24], Jacobs *et al.* arrive at the same complexity reduction—without throwing away the isometries altogether—through the use of a block orientation scheme. This scheme finds, for a pair of domain/range blocks, the isometries which will give a specific block orientation; say the orientation which puts the brightest block quadrant in the upper left corner and second brightest in the upper right one. When combined, these two isometries specify an isometry which maps domain block to a range block with the same orientation. This scheme is reported to be only very slightly suboptimal, compared with the full search of eight isometries.

C. Other Partitions, Other Block Shapes

We described in Sections III-A and V-C an encoding procedure for a two-level square partion of an image into 8×8 and 4×4 blocks, referred to, respectively, as parent and child blocks. It was observed by several researchers [10],

[24], [57] that by allowing more than one level of recursion in the partition, local "flatness" in images could be better exploited[7]. In [24], Jacobs *et al.* use a *quadtree partitioning method* to generate a partition which divides the image into nonoverlapping range blocks of sizes 32×32, 16×16, 8×8, and 4×4, with domain blocks always twice the respective size of the range blocks they attempt to match.

Domain blocks with sides tilted at a 45° angle with respect to the sides of a square (or rectangular) image were used by Jacobs *et al.* in [24]. These authors report that including such domain blocks results in higher encoding times, little change in compression, but increased coded image quality.

D. Fractal Block Coding and Other Block-Based Image Coding Techniques

Fractal coding is compatible with other, more traditional block coding techniques such as transform coding or vector quantization. This can be seen from (40) where these techniques can all be modeled as complex constant tranformations with no linear part, i.e., $\alpha_0 = 0$. For example, Δg could be either a DCT approximation of the block, or a best codeword picked in some image-independent codebook. Attempts to determine which coding techniques are preferable for which classes of blocks are reported in [30], [45]. In [45], the authors report that it might be advantageous to encode the "complex" regions[8] of an image with fractal block coding and the "smooth" regions with a block-DCT encoder. Somewhat different results are reported in [30], where we compared coding results obtained by using either fractal coding for texture and edge blocks or a simple memoryless VQ scheme, where codebooks of different sizes, designed with the Linde–Buzo–Gray (LBG) algorithm [34] were trained on image blocks of the same type as the blocks to encode. It appeared that for sharp-edge blocks, fractal coding performs better than VQ, but the results are very similar for texture blocks, with perhaps a slight advantage for VQ coding of texture block with an LBG codebook trained on texture blocks. Fractal block coding has also been compared to other more traditional block coding techniques such as adaptive DCT coding [24]. In [24] Jacobs *et al.* report that their best implementation of a fractal block coding system performs comparably, but not as well, as adaptive DCT.

Although the results are very preliminary, it is interesting to mention an extention of block-based fractal image coding to the encoding of digital video sequences by extending the theory to the three-dimensional case [8], [33]. In [8], Beaumont encodes slabs of four frames at a time, searching for matching domain blocks of size $12 \times 12 \times 12$. He uses a combination of three—dimensional fractal coding with a DPCM [29] prediction technique for the average luminance of range blocks. However, these authors report that perceptually annoying block effects occur. It is unclear whether the same coded image quality as in the still image case could be obtained at reasonably low bit rates, as affine block transformations may not efficiently exploit temporal correlation.

[7] This has also been observed by other researchers in different image coding contexts [11], [59], [60].

[8] Sharp edges and textures.

VII. Conclusion

We have described the design of digital image coding systems referred to as *Fractal Block Coders* which are based on a theory of iterated contractive image transformations. The preliminary design issues are to select an adaptive image partition made of nonoverlapping range cells, to select an interblock distortion measure, and to specify a class of contractive block transformations with quantized parameters. The encoding of an original image then consists of capturing the self-transformability of the original image by searching a global transformation pool for a transformation defined blockwise—a fractal code—under which the image is approximately invariant. A specific block-base fractal image coding system was presented as well as encoding/decoding results. The relationship between vector quantization and fractal block coding was analyzed.

The work of other researchers who either extended the theory, or optimized the original system design to improve the initial results of the author was then presented. The choices in the design a block-based fractal coding system affect all of the following: visual quality of coded images, compression, encoding complexity and speed, in a complex manner. Decoding complexity remains fairly low and stable for various system designs. Of course, the degree to which some of these issues are more important than others will have influence on what is considered an "optimal" fractal coding system. Specifically, the following has been observed. Quadtree-based image partitions should be used; they efficiently exploit the flat areas present in typical images. Different formulations of the block-matching problem of finding the best combination of domain and block transformation for a given range block can lead to efficient least squares optimizations of the parameters of an optimal block transformation, thereby considerably reducing the complexity of the initial full-search scheme. While keeping the requirement that the block transformations are affine, it is useful to both eliminate the initial strict contractivity requirement, as well as enlarge the pool of block transformations by allowing polynomial translation terms. The possible use of block shapes other than square was also evaluated, as well as that of combining fractal block coding with other more traditional block coding techniques.

Fractal block coding, as described in this paper, remains a fairly novel image coding technique. The interest that the publication of the initial work has generated, along with the number of publications which describe other implementations of fractal block coders for the encoding of images are exciting. Even though a complete reconciliation of the approaches taken by different authors may seem difficult at this point, the body of work provides useful insight to the design of a fractal block coder for images, given constraints of maximum complexity or bit rates. For

all its esoteric origins, fractal image coding performs quite comparably with more traditional image coding techniques; this despite the existence of a restricted class of complex-looking fractal objects which can be fully described by a very small set of parameters [4], [38]. Some might find this disappointing. We think that one main point of this work is that fractal block coding shows that there are ways to capture and exploit image redundancy in images, which are not used by more traditional image coding techniques. The property of piecewise self-similarity and its capture through the construction of contractive image transformations which leave original images approximately invariant provides a new scheme for the exploitation of image redundancy for image compression—this property is what makes fractal image coding work.

REFERENCES

[1] H. Abut, Ed., *Vector Quantization*. New York: IEEE Press, 1990.

[2] M. F. Barnsley and S. Demko, "Iterated function systems and the global construction of fractals," *Proc. Roy. Soc. London*, vol. A399, pp. 243–275, 1985.

[3] M. F. Barnsley, V. Ervin, D. Hardin, and J. Lancaster, "Solution of an inverse problem for fractals and other sets," *Proc. Nat. Acad. Sci.*, vol. 83, pp. 1975–1977, 1986.

[4] M. F. Barnsley, *Fractals Everywhere*. New York: Academic Press, 1988.

[5] M. F. Barnsley, J. H. Elton, and D. P. Hardin, "Recurrent iterated function systems," *Constructive Approximation*. Berlin, Germany: Springer-Verlag, 1989, pp. 3–31.

[6] M. F. Barnsley and A. Jacquin, "Application of recurrent iterated function systems to images," *Proc. SPIE*, , vol. 1001, pp. 122–131, 1988.

[7] M. F. Barnsley, A. Jacquin, F. Malassenet, and L. Reuter, "Harnessing chaos for image synthesis," in *Proc. SIGGRAPH '88*, 1988, pp. 131–140.

[8] J. M. Beaumont, "Image data compression using fractal techniques," *BT Technol. J.*, vol. 9, no. 4, pp. 93–109, Oct. 1991.

[9] T. J. Bedford, F. M. Dekking, and M. S. Keane, "Fractal image coding techniques and contraction operators," Delft Univ. Technol., Internal Rep., 1992.

[10] T. J. Bedford, F. M. Dekking, M. Breeuwer, M. S. Keane, and D. van Schooneveld, "Fractal coding of monochrome images," Delft Univ. Technol., Internal Rep., 1992.

[11] C. Chen, "Adaptive transform coding via quadtree-based variable blocksize DCT," in *Proc. ICASSP-89*, May 1989, pp. 1854–1857.

[12] G. Cherbit, *Fractals, Dimensions non Entières et Applications*. Paris, France: Masson, 1987.

[13] N. Dunford and J. T. Schwartz, *Linear Operators*. New York: Wiley, 1966.

[14] K. Falconer, *The Geometry of Fractal Sets*. London, UK: Cambridge Univ. Press, 1985.

[15] ——, *Fractal Geometry, Mathematical Foundations and Applications*. New York: Wiley, 1990.

[16] Y. Fisher, E. W. Jacobs, and R. D. Boss, "Iterated transform image compression," NOSC Tech. Rep. TR-1408, Naval Oceans Systems Center, San Diego, CA, 1991.

[17] A. Fournier, D. Fussell, and L. Carpenter, "Computer rendering of stochastic models," *Commun. ACM*, vol. 25, pp. 371–384, 1982.

[18] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Dordrecht, The Netherlands: Kluwer, 1992.

[19] J. Gleick, *Chaos, Making of a New Science*. New York: Vicking, 1987.

[20] R. C. Gonzalez and P. Wintz, *Digital Image Processing*. Reading, MA: Addison-Wesley 1977.

[21] R. M. Gray, "Vector quantization," *IEEE ASSP Mag.*, Apr. 1984.

[22] J. C. Hart, D. J. Sandin, and L. H. Kauffman, "Ray tracing deterministic 3-D fractals," *Comput. Graph.*, vol. 23, no.3, pp. 91–100, 1989.

[23] J. C. Hart and F. K. Musgrave (Co-Chairs), "Fractal modeling in 3D computer graphics and imaging," in *SIGGRAPH '91*, course notes, 1991.

[24] E. W. Jacobs, Y. Fisher, and R. D. Boss, "Image compression: A study of the iterated transform method," *Signal Process.*, vol. 29, no. 3, pp. 251–263, Dec. 1992.

[25] A. E. Jacquin, "A fractal theory of iterated Markov operators with applications to digital image coding," Ph.D. dissertation, Georgia Tech., Atlanta, 1989.

[26] ——, "A novel fractal block-coding technique for digital images," in *Proc. ICASSP-90*, 1990, pp. 2225–2228.

[27] ——, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Trans. Image Process.*, vol. 1, no. 1, pp. 18–30, Jan. 1992.

[28] A. K. Jain, "Image data compression: A review," *Proc. IEEE*, vol. 69, Mar. 1981.

[29] N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[30] T. Laurençot and A. Jacquin, "Hybrid image block coders incorporating fractal coding and vector quantization, with a robust classification scheme," AT&T Tech. Memo., Feb. 1992.

[31] S. Lepsøy, "Attractor image compression—fast algorithms and comparisons to related techniques," Ph.D dissertation, The Norwegian Institute of Technology, Trondheim, Norway.

[32] S. Lepsøy, G. E. Øien, and T. A. Ramstad, "Attractor image compression with a fast non-iterative decoding algorithm," in *Proc. ICASSP-93*, 1993, pp. 337–340.

[33] H. Li, M. Novak, and R. Forchheimer, "A fractal-based image sequence compression scheme," Preprint, 1993.

[34] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.

[35] L. Lundheim, "An approach to fractal coding of one-dimensional signals," in *Proc. KOMPRESJON-89* (Oslo, Norway), 1989.

[36] ——, "Fractal signal modelling for source coding," Ph.D dissertation, The Norwegian Institute of Technology, Trondheim, Norway, Sept. 1992.

[37] F. J. Malassenet, "Texture coding using a pyramid decomposition," in *Proc. ICASSP-93*, 1993, vol. V, pp. 353–356.

[38] B. Mandelbrot, *The Fractal Geometry of Nature*. San Francisco, CA: Freeman, 1982.

[39] D. M. Monro and F. Dudbridge, "Fractal approximation of image blocks," in *Proc. ICASSP-92*, 1992, vol. III, pp. 485–488.

[40] D. M. Monro and F. Dudbridge, "Fractal block coding of images," *Electron. Lett.*, vol. 28, pp. 1053–1055, May 1992.

[41] D. M. Monro, "Generalized fractal transforms: Complexity issues," in *Proc. Data Compression Conf.*, Mar.–Apr. 1993, IEEE Computer Soc. Press, pp. 254–261.

[42] A. N. Netravali and B. G. Haskell, *Digital Pictures: Representation and Compression*. New York: Plenum, 1989.

[43] G. E. Øien, S. Lepsøy, and T. A. Ramstad, "An inner product space approach to image coding by contractive transformations," in *Proc. ICASSP-91*, 1991, pp. 2773–2776.

[44] ——, "Reducing the complexity of a fractal-based image coder," presented at EUSIPCO-92.

[45] G. E. Øien, A. Sollid, and T. A. Ramstad, "Hybrid image compression combining transform coding and attractor coding," presented at NOBIM-92.

[46] G. E. Øien, "L_2-optimal attractor image coding with fast decoder convergence," Ph.D. dissertation, The Norwegian Institute of Technology, Trondheim, Norway, 1993.

[47] H-O. Peitgen and P. H. Richter, *The Beauty of Fractals*. Berlin: Springer, 1986.

[48] H-O. Peitgen and D. Saupe, *The Science of Fractal Images*. New York: Springer-Verlag, 1988.

[49] H-O. Peitgen, H. Jürgens, and D. Saupe, *Chaos and Fractals*. New York: Springer-Verlag, 1992.

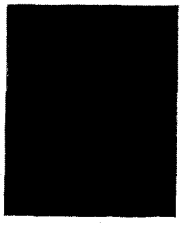
[50] A. P. Pentland, "Fractal-based descriptions of natural scenes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, no. 6, 1984.

[51] ——, "Fractal surface models for communications about terrain," *SPIE Visual Comun. Image Process. II*, vol. 845, 1987.

[52] M. Rabbani and P. W. Jones, *Digital Image Compression Techniques*, vol. TT7. Bellingham, WA: SPIE Optical Eng. Press, 1991.

[53] B. Ramamurthi and A. Gersho, "Classified vector quantization of images," *IEEE Trans. Commun.*, vol. COM-34, Nov. 1986.

[54] L. Hodges-Reuter, "Rendering and magnification of fractals using iterated function systems," Ph.D. dissertation, Georgia Inst. Technol., Atlanta, 1987.
[55] R. Rinaldo and A. Zakhor, "Fractal approximation of images," in *Proc. Data Compression Conf.*, Mar.–Apr. 1993, p. 451.
[56] H. Samet, "The quad-tree and related hierarchical data structures," *ACM Comput. Surv.*, vol. 16, no. 2, pp. 188–260, June 1984.
[57] D. van Schooneveld, "Fractal coding of monochrome images," Delft Univ. Technol., Internal Rep., Dec. 1991.
[58] M. C. Stein, "Fractal image models and object detection," *SPIE Visual Commun. Image Process. II*, vol. 845, 1987.
[59] J. Vaisey and A. Gersho, "Variable block-size image coding," in *Proc. ICASSP-87*, 1987, pp. 1051–1054.
[60] ——, "Image compression with variable block size segmentation," *IEEE Trans. Signal Process.*, vol. 40, no. 8, pp. 2040–2060, Aug. 1992.



Arnaud Jacquin (Member, IEEE) was born in Reims, France, in 1964. He received the Diplôme d'Ingénieur degree in electrical engineering from Ecole Supérieure d'Electricité, Gif-sur-Yvette, France, in 1986. He received the M.S. degree in electrical engineering in 1987, and the Ph.D. degree in mathematics in 1989, both from the Georgia Institute of Technology, Atlanta.

Since 1990, he has been a Member of the Technical Staff at AT&T Bell Laboratories, Murray Hill, NJ, in the Signal Processing Research Department, where he has been engaged in research on low bit rate video coding for teleconferencing/videotelephony. His research interests are in the areas of image and video compression, fractals, computer vision, and computer graphics.

Dr. Jacquin is the recipient of the IEEE Signal Processing Society's 1993 Senior Award in the area of Multidimensional Signal Processing, for his paper "Image coding based on a fractal theory of iterated contractive image transformations," which was published in the January 1992 issue of the IEEE TRANSACTIONS ON IMAGE PROCESSING.