A NOVEL FRACTAL BLOCK-CODING TECHNIQUE FOR DIGITAL IMAGES

Arnaud E. Jacquin

School of Mathematics
Georgia Institute of Technology
Atlanta, GA 30332

Abstract

A novel approach to digital image coding, rooted in Iterated Transformation Theory, and referred to as ITT-based coding, is proposed. It is a *fractal* block-coding method which relies on the assumption that image redundancy can be efficiently exploited through *block self-transformability.*

The coding-decoding system is based on the construction, for any given original image to encode, of an image transformation of a special kind which, when iterated on any initial image, produces a sequence of images that converges to a fractal approximation of the original. The requirements on the transformation are that (i) it is *contractive* in the metric space of images endowed with the L_2 metric, (ii) it leaves the original image approximately invariant, and (iii) its *complexity* is smaller than that of the original image. Our fully automated ITT-based system has comparable performance, in terms of SNR and bit rate, to state-of-the-art vector quantizers, with which it shares some features.

I INTRODUCTION

The past few years have seen the rise of various block-coding methods for the compression of digital images. Modern techniques essentially revolve around the areas of transform coding [7] and vector quantization [2,6].

In this paper, we propose a novel image block-coding technique. It is rooted in a mathematical theory of Iterated Transformations [3,4]. The transformations considered here are discrete, contractive image transformations defined blockwise.

Given an original image, the encoding procedure is to find among a class of contractive transformations specified *a priori*, one which leaves the original image approximately invariant. We call such a transformation a fractal code. It has the fundamental property that, when iterated on any initial image, it produces a sequence of images which converges to a fractal image—the decoded image—visually close to the original.

In section II we propose a class of transformations. In section III we present an automated ITT-based coding procedure. In section IV we address the decoding of an image from a fractal code, and show how to compute bit rates. In section V we present coding simulations and compare our technique with vector quantization, on a theoretical level.

II A CLASS OF DISCRETE IMAGE TRANSFORMATIONS

In this section, we define a class of discrete image transformations which will be used to *represent* or *encode* digital images. These transformations are defined blockwise, consistently with a partition of the image support, and are contractive with respect to the L_2 metric.

A. Image partitions

The square support of an $r \times r$ digital image is partitioned into non-overlapping square range cells[1] of two different sizes; thus forming a two-level square partition. The larger cells—of size $B \times B$—are referred to as (range) parent cells; the smaller ones—of size $B/2 \times B/2$—as (range) child cells. A parent cell can be split into four non-overlapping child cells. The actual construction of a partition, given an original image to encode, is discussed in section III. All along this paper, the image partition will be denoted $\{B_i\}_{i \in C}$, where C is the set of all range blocks indices.

B. Distortion measure

Let $S(i_0,j_0,B)$ denote the square cell of size $B \times B$, with bottom left corner at the intersection of image row i_0 and image column j_0. Let μ be an original $r \times r$ image, and $\tilde{\mu}$ be an approximation of μ. Let $\mu(S), \tilde{\mu}(S)$ denote their restrictions to the cell $S(i_0,j_0,B)$. The L_2 or root-mean-square (rms) distortion between the image blocks $\mu(S)$ and $\tilde{\mu}(S)$ is defined as the square root of the sum, over the cell, of the squared differences of pixel values, i.e.

$$d_{L_2}(\mu(S),\nu(S)) = \Big(\sum_S (\mu_{i_0+i,j_0+j} - \nu_{i_0+i,j_0+j})^2\Big)^{\frac{1}{2}}.$$

The signal-to-noise ratio (SNR) is usually defined by:

$$\mathrm{SNR} = 10\log_{10}\Big(\frac{dr(\mu)^2}{d_{L_2}(\mu,\tilde{\mu})^2/r^2}\Big),$$

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

C. Discrete image transformations

The general form of our image transformations is a sum of elementary block transformations:

$$\tau = \textstyle\sum_{i \in C} \tau_i = \sum_{i \in C} \mathcal{T}_i \circ \mathcal{S}_i,$$

where $\mathcal{S}_i$ is a spatial contraction from a domain cell D_i to the range cell B_i, and $\mathcal{T}_i$ is a transformation which processes image blocks supported on B_i.

- **Spatial contraction** $\mathcal{S}_i$

We describe the action of a $D:B$ ($D = 2B$) discrete spatial contraction operator $\mathcal{S}$ which maps a square domain cell $D_i = S(i_d,j_d,D)$ to a square range cell $B_i = S(i_r,j_r,B)$. The pixel values of the contracted image block on the range cell B_i are average values of four neighboring pixels in the domain block.

$$(\mathcal{S}\mu)_{i_r+i,j_r+j} = (\mu_{I(i),J(j)} + \mu_{I(i)+1,J(j)} + \mu_{I(i),J(j)+1} + \mu_{I(i)+1,J(j)+1})/4, \text{ for all } i,j \in \{0,\ldots,B-1\},$$

where the two index functions I and J are defined by, $I(i) = i_d + 2i$, and $J(j) = j_d + 2j$.

[1]The term *cell* is used to denote a square subset of the image support. The term *block* is used to denote the restriction of an image to a cell.

• **Block processing $\mathcal{T}_i$**

A few simple linear processing operators are listed below.

i) Absorption at gray level g_0: $\forall i,j \in I,\ (\theta\mu)_{i,j} = g_0$.

ii) Translation by Δg: $\forall i,j \in I,\ (\tau\mu)_{i,j} = \mu_{i,j} + \Delta g$.

iii) Gray level scaling by $\alpha \in [0,1]$: $\forall i,j \in I,\ (\sigma\mu)_{i,j} = \alpha\mu_{i,j}$.

The following transformations do not modify pixel values; they simply *shuffle* pixels within a range block, in a deterministic way—we call them isometries. We give below a list of the eight canonical isometries of a square block.

ι_0: Identity,
ι_1: Orthogonal reflection about mid-vertical axis of block,
ι_2: Orthogonal reflection about mid-horizontal axis of block,
ι_3: Orthogonal reflection about first diagonal (i = j) of block,
ι_4: Orthogonal reflection about second diagonal of block,
ι_5: Rotation around center of block, through +90°,
ι_6: Rotation around center of block, through +180°,
ι_7: Rotation around center of block, through −90°.
Examples of more complex transformations can be found in [5].

The L_2-contractivities of the discrete block transformations described above can be easily computed. They are given in table 1.

Massic transformation	L_2-contractivity
Absorption	$s = 0$
Translation	$s = 1$
Gray level scaling by α	$s = \alpha$
Isometries $\{\iota_n\}_{0 \le n \le 7}$	$s = 1$

Table 1. L_2-contractivities of block processing transformations.

The numerical parameters of the transformations will be quantized. The description of a block transformation should be kept as simple as possible, in order to obtain low bit rates for ITT-based encoding, as will be seen in section IV.*B*.

III CONSTRUCTION OF FRACTAL CODES FOR IMAGES

The *inverse problem* of Iterated Transformation Theory [1,3,4,5] applied to digital images is an *image coding problem.* Given an original r×r digital image μ to encode, it consists of finding among the class of image transformations defined in section II, a transformation τ which leaves μ approximately invariant, i.e. which minimizes the distortion:

$$d_{L_2}(\mu, \tau(\mu)).$$

An image μ for which a transformation τ exists, such that this distortion is "small", is said to be approximately **block self-transformable.** The image $\tau(\mu)$ is then called a **collage** of μ.

Since our image transformations have the form:

$$\tau = \sum_{i \in C} \tau_i, \text{ with } \tau_i = \mathcal{T}_i \circ \mathcal{I}_i,$$

the problem is to find, for every range block B_i of size b×b[2] in the image partition, a domain block D_i and a block processing transformation $\mathcal{T}_i$, such that the distortion:

$$d_{L_2}(B_i, \mathcal{T}_i \circ \mathcal{I}_i(D_i)) \text{ is mimimum.}$$

• **Pools of domain blocks**

The maximal domain pool for a range block size b×b consists of ***all*** blocks in the original image to encode of size 2b×2b. It is typically very large, but can be classified according to block geometry. Using a thorough study of image block classification by Ramamurthi and Gersho [8], we create four classes for **shade blocks**, simple and **mixed edge blocks**, and **midrange blocks**. We denote these classes of domain blocks $\mathcal{D}_s$, $\mathcal{D}_{se}$, $\mathcal{D}_{me}$, and $\mathcal{D}_m$, respectively.

• **Selection of the block processing $\mathcal{T}_i$**

Every element D_k of the domain pool of the same type as the block B_i is scanned. The analysis of a pair (B_i, D_k) determines the selection of an optimal block processing transformation $\mathcal{T}_i$, according to a procedure described in [3,5]. The optimal transformed domain block $\mathcal{T}_i \circ \mathcal{I}_i(D_i)$ is called the **matching block**[3] for B_i.

• **Construction of the image partition**

The coding of a range parent block is performed in the following way. First, an optimal **parent transformation**[4] τ_i is selected according to the procedure mentioned above. The parent matching block is then split into four children, and distortion measures between these and the original child blocks are computed. For every child block such that the distortion is too high, a **child transformation** $\tau^{p,q}$ is constructed and is "appended" to the parent transformation. There is therefore sixteen possible coding configurations of a parent block, which can be represented with $I_c = 4$ bits. Details about this image-dependent construction can be found in [3,5].

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

$$\{(\tau_i, \tau_i^{1,1}, \tau_i^{1,2}, \tau_i^{2,1}, \tau_i^{2,2}), i \in C\}.$$

The structure of a fractal code is illustrated in figure 1, where the arrows indicate block transformations from domain block to range block.

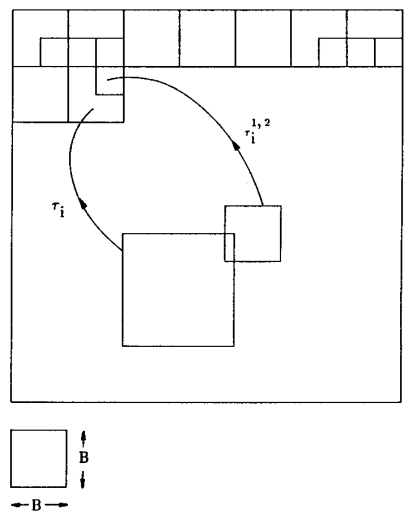

Fig. 1. Description of the structure of a fractal image code.

[2] b is used to denote either B or B/2, according to the nature of B_i, parent or child.

[3] In effect, we select a matching block from a virtual codebook made of contracted and processed image blocks. The analogy with the notion of codebook used in vector quantization will be discussed in section V.*B*.

[4] We distinguish **parent transformations**, which map to image blocks of size B×B (parent blocks), from **child transformations**, which map to blocks of size B/2×B/2 (child blocks).

IV DECODING FRACTAL CODES

A. Image reconstruction

The decoding scheme simply consists in iterating a code τ on any initial image μ_0, until convergence to a final decoded image is observed[5]. The sequence of images:

$$\{\mu_n = \tau^n(\mu_0)\}_{n=0}^{\infty},$$

is called a **fractal reconstruction sequence**. The mapping of an image under a fractal code is done sequentially. For each cell index i, the transformation τ_i is applied to the image block over the domain D_i, and mapped onto the range cell B_i.

In figure 2, are displayed the first eight iterations of a fractal code for the "lena" image, applied to an initial black image. The values of the SNR between the original "lena" image and successive terms of the reconstruction sequence are listed in table 2. Convergence of the sequence of images is obtained, within .2 dB of accuracy, at the eighth iteration.

it. no	1	2	3	4	5	6	7	8	9	10
SNR	5.9	19.5	22.8	25.2	26.4	27.0	27.3	27.5	27.6	27.6

Table 2. Values of the SNR between the original "lena" image and successiveterms of the reconstruction sequence.

B. Computation of bit rates

The full description of a fractal code τ, in view of its storage or transmission, is the key to the evaluation of bit rates. It depends on: (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.

Three distinct types of parameterized block transformations are used, depending on the type of the underlying range block: shade, midrange, or edge. The information in bits needed to represent a block transformation of each type is computed from the encoding characteristics listed in table 4, and is presented in table 3.

Block type	Parameters	Information in bits
shade	g_i	6
		$I_s = 6$ bits
midrange	D_i (domain)	$7 + 7 = 14$
	α_i	2
	Δg_i	7
		$I_m = 23$ bits
edge	D_i	$7 + 7 = 14$
	α_i	3
	Δg_i	7
	ι_{n_i}	3
		$I_e = 27$ bits

Table 3. Information in bits for the representation of block transformations.

Let N_s, N_m, and N_e denote the total numbers—parents and children—of range shade blocks, midrange blocks, and edge blocks in the original image. The bit rate is equal to:

$$\frac{N_p I_c + N_s I_s + N_m I_m + N_e I_e}{N_p B^2} \text{ bpp.}$$

where $N_p = \frac{r}{B}$ is the total number of parent blocks in the image.

[5]The convergence of any such iterated sequence of images to the same stable fractal image is a consequence of the contractivity of the fractal code. A thorough theoretical justification of this property can be found in [3,4].

V CODING RESULTS

A. Coding simulation

The characteristics of the ITT-based system used for the encoding of the 256×256, 6 bpp, "lena" image are given in table 4. The original image was initially split into four 128×128 subimages, which were encoded independently. The decoded image—a fractal approximation of the original, is displayed in figure 3.

The performance of the encoding system, in terms of bit rate and fidelity of the decoded image to the original, for the particular block statistics specific to "lena", is also given in table 4. Image "textures" are well preserved although some finely textured areas, such as the turban around lena's hat, are "smoothed out" by the encoding process. Sharp, contrasted contours, whether they are smooth or rugged, are very accurately preserved. Some *blocking* artefact is visible, as is expected for the type of memoryless block-encoding method used here. However, the reconstruction is free of edge degradation by "staircase effect". This is especially clear in the magnification on lena's shoulder, shown in figure 4[6], where the outline of the shoulder is perfectly smooth.

Encoding specifications	
Partition	
Range blocks:	8×8 (parent), 4×4 (child)
Domain pool	
Domain blocks:	16×16 (parent), 8×8 (child)
Classification:	shade (s), midrange (m), simple edge (se), mixed edge (me)
Transformation pool	
Shade block:	Absorption at $g \in \{gl_{min}, \ldots, gl_{max}\}$
Midrange block:	Gray level scaling by $\alpha \in \{.7, \ldots, 1.0\}$
	Translation by $\Delta g \in \{-gl_{max}, \ldots, gl_{max}\}$
Edge block:	Gray level scaling by $\alpha \in \{.2, \ldots, .9\}$
	Translation by $\Delta g \in \{-gl_{max}, \ldots, gl_{max}\}$
	Isometries $\{\iota_n\}_{0 \leq n \leq 7}$
System performance	
Block statistics:	s: 11.0 %, m: 30.0 %, e: 59.0 %
Bit rate:	.68 bpp
SNR:	27.7 dB

Table 4. Design specifications and performance of the ITT-based system used for the encoding of "lena".

B. Analogies with vector quantization

The practical implementation of the ITT-based coding system designed in this research work is related to **vector quantization** [2,6,8]. In table 5, we give a comparative list of features shared by vector quantization and ITT-based coding.

It is important to note that, with our technique, domain image blocks are not needed by the decoder. The virtual codebook of contracted and processed domain blocks is never transmitted. It is used only during the encoding phase, for the construction of the fractal code. The iterative reconstruction of an image is very specific of ITT-based decoders.

The fractal method described in this research work shows very strong promise as a completely novel approach to digital image coding. Our ITT-based coding-decoding system yields encouraging preliminary results, which can certainly be further improved.

[6]This image was obtained by reconstructing the bottom-left subimage of the code, with an initial image of size 256×256—twice the size of the subimage which is used in the reconstruction of a complete "lena" image. This illustrates the fact that a reconstruction sequence is not attached to any specific image size. Details about the magnification of images decoded from fractal codes can be found in [4,5].

Fig. 2. First eight iterations of a code for "lena", applied to an initial uniformly black image.

VQ	ITT–based coding
• Partition Partitioning of image in square blocks.	Partitioning of image in range blocks.
• Codebook Training set of images. Codebook design. Off–line transmission of codebook.	• Virtual codebook Contracted and processed domain blocks extracted from original image itself. Trimming of domain pool could be done. *No transmission of domain blocks.*
• Block matching Selection of a block distortion measure. Use of the notion of block classification.	• Encoding of range blocks Same type of block matching in virtual codebook.
• Decoding Image code: list of addresses of blocks in the codebook. Direct reconstruction by look–up table.	• Reconstruction, decoding Fractal code: list of block transformations consistent with an image partition. Iterative reconstruction.

Table 5. Analogies between vector quantization and ITT–based encoding.

Fig. 3. Decoded fractal "lena" image.

Fig. 4. Magnification on shoulder of decoded "lena".

REFERENCES

[1] Barnsley, M.F., Ervin, V., Hardin, D., Lancaster, J., "*Solution of an inverse problem for fractals and other sets*", Proc. Natl. Acad. Sci. USA, 83 (1986).

[2] Gray, R.M., "*vector quantization*", IEEE ASSP Magazine, April 1984.

[3] Jacquin, A.E., *A fractal theory of iterated Markov operators with applications to digital image coding*, Ph.D. Thesis, Georgia Tech, 1989.

[4] Jacquin, A.E., "*Image coding based on a fractal theory of iterated contractive Markov operators, part I: Theoretical foundation*", Georgia Tech Preprint 091389–016, 1989, submitted to Trans. on ASSP.

[5] Jacquin, A.E., "*Image coding based on a fractal theory of iterated contractive Markov operators, part II: Construction of fractal codes for digital images*", Georgia Tech Preprint 091389–017, 1989, submitted to Trans. on ASSP.

[6] Nasrabadi, N.M., King, R.A., "*Image coding using VQ: A review*", IEEE Trans. on Communications, vol. 36, no. 8, Aug. 1988.

[7] Netravali, A.N., Limb, J.O.,"*Picture coding: A review*", Proc. IEEE vol. 68, no. 3, March 1980.

[8] Ramamurthi, B., Gersho, A., "*Classified vector quantization of images*", IEEE Trans. on Communications, vol. 34, no. 11, Nov. 1986.