On the Inverse Problem of Fractal Compression

Hannes Hartenstein[1], Matthias Ruhl[2], Dietmar Saupe[3], and Edward R. Vrscay[4,*]

[1] Computer & Communication Research Lab, NEC Europe Ltd., Heidelberg, Germany, E-mail: Hannes.Hartenstein@ccrle.nec.de

[2] Laboratory of Computer Science, Massachusetts Institute of Technology, USA E-mail: ruhl@mit.edu

[3] Institut für Informatik, Universität Leipzig, Germany E-mail: saupe@informatik.uni-leipzig.de

[4] Department of Applied Mathematics, University of Waterloo, Canada E-mail: ervrscay@links.uwaterloo.ca

Abstract. The inverse problem of fractal compression amounts to determining a contractive operator such that the corresponding fixed point approximates a given target function. The standard method based on the *collage coding* strategy is known to represent a suboptimal method. Why does one not search for optimal fractal codes? We will prove that optimal fractal coding, when considered as a discrete optimization problem, constitutes an NP-hard problem, i.e., it cannot be solved in a practical amount of time. Nevertheless, when the fractal code parameters are allowed to vary continuously, we show that one is able to improve on collage coding by fine-tuning some of the fractal code parameters with the help of differentiable methods. The differentiability of the attractor as a function of its luminance parameters is established. We also comment on the approximating behavior of collage coding, state a lower bound for the optimal attractor error, and outline an annealing scheme for improved fractal coding.

1 Introduction

Fractal compression seeks to approximate a target function f with a function $\bar{f}_p$ which is the fixed point, or *attractor*, of a 'simple' contractive operator T_p that acts on a suitable metric space $(\mathcal{F}, d_{\mathcal{F}})$ of functions. The parameter vector p (also called the *fractal code*) that defines T_p (also called *fractal transform operator*) is then used as a (lossy) representation of the target function f. The fixed point $\bar{f}_p$ is generated by iterating the operator T_p on an arbitrary function of the space $\mathcal{F}$; this is the decoding step. The encoding problem of fractal compression lies in finding in a suitable class of operators the one whose corresponding fixed point gives the best approximation of the target function. Of course, the class of operators considered for fractal coding purposes has to be constrained to 'simple' operators that can be coded compactly in order to lead to data compression. The encoding problem is

[*] Project: Partitioned Iterated Function Systems. The Inverse Problem of Image Coding and the Complexity of Associated Dynamical Systems (Dietmar Saupe)

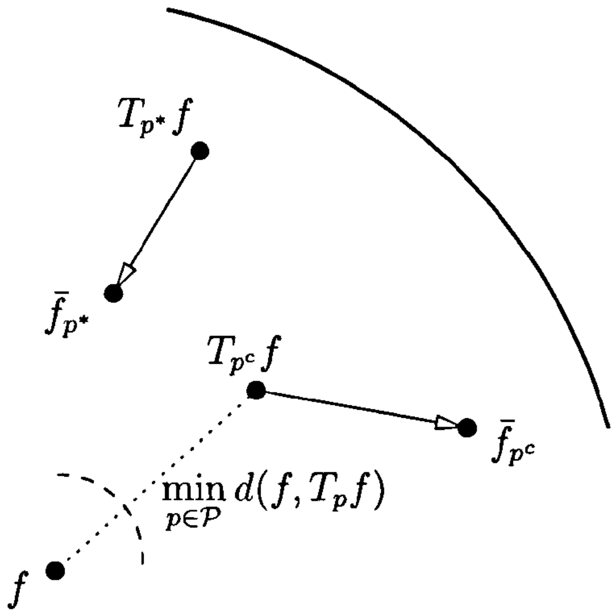


Fig. 1. Schematic presentation of the relationship between the target f, an optimal collage $T_{p^c}f$, the fixed point $\bar{f}_{p^c} = T_{p^c}\bar{f}_{p^c}$ corresponding to T_{p^c}, as well as the collage $T_{p^*}f$ of an optimal fractal code and the corresponding optimal fixed point $\bar{f}_{p^*} = T_{p^*}\bar{f}_{p^*}$. The boldfaced arc indicates the upper bound on the feasible attractor error provided by the collage theorem while the dotted arc indicates a lower bound on the optimal attractor error. This lower bound is given in Section 5. Clearly, a collage optimal fractal code does not in general coincide with an optimal fractal code.

also called the inverse problem of fractal compression since it involves the determination of 'causes', i.e., the determination of the operator parameters, based on a desired 'effect', i.e., the desired fixed point.

In practice, fractal coding algorithms rely upon the method of *collage coding.* Given a target function f and a suitable parameter space $\mathcal{P}$ one determines a fractal transform operator $T_{p^c}, p^c \in \mathcal{P}$, that minimizes the *collage error* $d_{\mathcal{F}}(f, Tf)$. This procedure is motivated by the *collage theorem* [1,3], a corollary of the contraction mapping principle. The collage theorem states that the attractor error $d_{\mathcal{F}}(f, \bar{f}_{p^c})$ is bounded from above by a multiple of the collage error $d_{\mathcal{F}}(f, T_{p^c}f)$. Thus, with the collage coding method one minimizes a bound on the actual attractor error. However, with this approach one generally does not find an *optimal* fractal code for the target function f, i.e., a fractal code $p^* \in \mathcal{P}$ such that $d_{\mathcal{F}}(f, \bar{f}_{p^*}) = \min_{p\in\mathcal{P}} d_{\mathcal{F}}(f, \bar{f}_p)$ (see Figure 1).

It is, therefore, natural to address the question: why does one not search for an optimal fractal code? In this paper we will show the following result: *Optimal fractal coding is NP-hard.* Thus, fractal coding —when considered as an optimization problem— represents an intractable problem, i.e., it is the computational complexity that prevents us from determining optimal fractal codes.

Given that optimal fractal coding is intractable, can one at least improve upon collage coding or is collage coding essentially the best one can do? We will show that one is able to improve on collage coding by fine-tuning some of the fractal code parameters with the help of differentiable methods.

The above short outline of the results indicates that we tackle the inverse problem of fractal compression from two different directions using two different mathematical methodologies. For the NP-hardness proof we consider fractal coding as a discrete optimization problem, whereas for the improvements over collage coding some of the parameters are assumed to be continuous. Since it is easier to state the discrete problem by reference to the 'continuous' problem, the first part of the paper will deal with the question of how to improve on collage coding with the use of differentiable methods, while the second part presents the NP-hardness result —in contrast to the 'logical' order of arguments. This paper summarizes the main results from our conference publications [27,28].

The use of contractive transforms and their corresponding attractors for the compression of signals and images was proposed by Barnsley and Jacquin in the late 1980s [2,19]. Before the birth of fractal compression and without technical applications in mind, Williams [29] and Hutchinson [18] had published mathematical studies of compositions of contractions and iterated function system. During the last 10 years about 400 papers were published in the field of fractal compression, as well as four books [4,10,20,11]. Several studies have attempted to find attractor functions $\bar{f}$ that are better approximations to a target f than the "collage attractors" $\bar{f}_{p^c}$. Indeed, these studies have typically employed the collage attractor $\bar{f}_{p^c}$ as a starting point. For example, Barthel [5] and then Lu [20] have devised "annealing schemes" that produce sequences of attractors $\bar{f}^{(n)}$ that are then used to "collage" the target f. The sequences $\bar{f}^{(n)}$ are observed to provide better approximations to the target. However, there is still no rigorous theoretical basis for this method. On the other hand, Dudbridge and Fisher [9], using the Nelder-Mead simplex algorithm, searched the fractal code space $\mathcal{P}$ in the vicinity of the collage attractor to locate (local) minima of the approximation error $d_{\mathcal{F}}(f, \bar{f})$. Their method was applied to a restricted class of (separable) fractal transforms, in which four 4×4 pixel range blocks shared a common domain block [22]. Withers [30] has derived differentiability properties of Iterated Functions Systems with probabilities whose attractors model graphs of 1D functions. Newton's method is used to compute parameters.

The paper is organized as follows: below notations and basic definitions are introduced. In Section 3 the differentiability of the attractor functions with respect to the luminance parameters is proven and results obtained by using gradient methods are presented. In Section 4, the problem of optimal fractal coding is stated as an combinatorial problem, and the computational complexity of this problem is analyzed. Further results are surveyed in Section 5.

2 Mathematical and Notational Preliminaries

What is the form of a fractal transform operator T? Let (X, d) denote the support or *base space*, assumed to be a metric space, e.g. $X = [0, 1)$ or $X = [0, 1)^2$. Let $\mathcal{F}(X) = \{f : X \to \mathbb{R}\}$ denote a suitable complete space of functions with metric $d_{\mathcal{F}}$. Now let $R_k \subset X, k = 1, 2, \dots, n_{\mathcal{R}}$ denote a set of *range blocks* that partition X, i.e., (1) $\bigcup_{k=1}^{n_{\mathcal{R}}} R_k = X$ and (2) $R_i \cap R_j = \emptyset$ for $i \neq j$. With each range block are associated the following:

1. a *domain block* $D_k \subset X$ and a one-to-one contraction map $w_k : D_k \to R_k$ with a contraction factor $c_k \in [0, 1)$.
2. an affine map $\phi : \mathbb{R} \to \mathbb{R}, \phi_k(t) = s_k t + o_k$, where $s_k, o_k \in \mathbb{R}$.

In the language of [13], the above ingredients comprise an (affine) $n_{\mathcal{R}}$-map Iterated Function System with Grey Level Maps (IFSM). The fractal transform operator $T : \mathcal{F}(X) \to \mathcal{F}(X)$ associated with such a (nonoverlapping) IFSM is defined as follows. Given a function $f \in \mathcal{F}(X)$ then for all $x \in R_k$, $k = 1, 2, \dots, N$,

$$\begin{aligned}(Tf)(x) &= \phi_k(f(w_k^{-1}(x)) \\ &= s_k f(w_k^{-1}(x)) + o_k. \end{aligned} \tag{1}$$

The maps w_k incorporate some form of *self-reference*. When considering the function values $f(x)$ as *luminance values* one can view the parameters s_k and o_k as control parameters for *contrast* and *brightness* (s stands for *scaling factor*, o for *offset*). They are also called *luminance parameters*. Figure 2 illustrates a fractal transform operator for the case of image coding.

It is well known that if $|s_k| < 1, 1 \leq k \leq n_{\mathcal{R}}$, then the operator T is contractive in the complete metric space of functions $\mathcal{L}^\infty(X)$. In the complete metric space of functions $\mathcal{L}^2(X)$, a straightforward calculation shows that

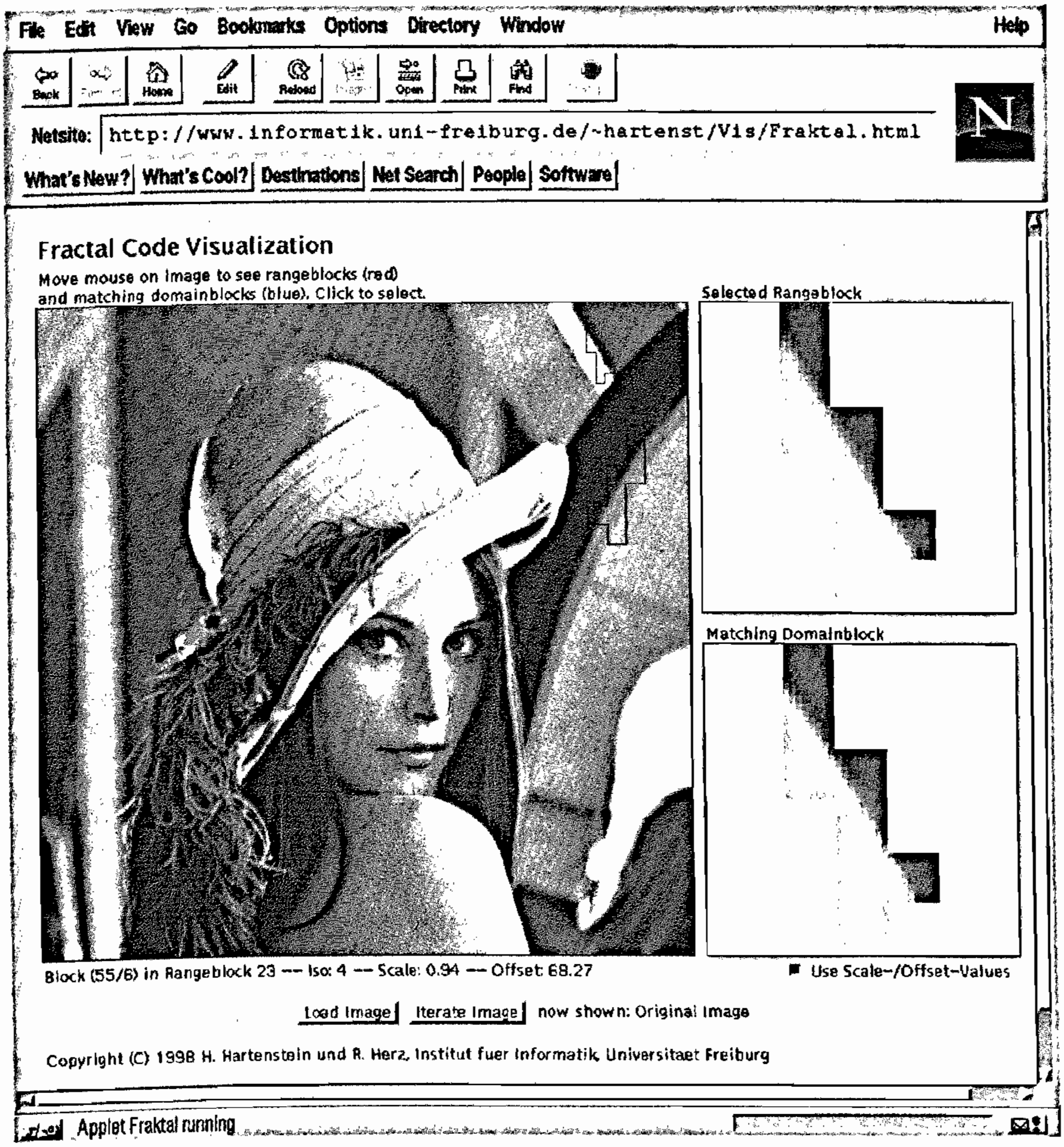
$$\|Tf_1 - Tf_2\|_2 \leq C\|f_1 - f_2\|_2, \quad \forall f_1, f_2 \in \mathcal{L}^2(X),$$

where

$$C = \sum_{k=1}^{n_{\mathcal{R}}} c_k |s_k|. \tag{2}$$

Therefore, the condition $C < 1$ is sufficient (but not necessary) for contractivity of T in $\mathcal{L}^2(X)$. An example of the iterative application of an contractive fractal transform operator is given in Figure 3.

In the first part of this paper, we assume that one is given a target function, a range partition as well as a range-domain assignment. Then we examine a systematic method to perform attractor optimization using the partial derivatives of attractor functions with respect to the luminance parameters, $\partial\bar{f}_p/\partial s_k, \partial\bar{f}_p/\partial o_k, k = 1, 2, \dots, n_{\mathcal{R}}$. To this end, we use $\mathcal{F}(X) = \mathcal{L}^2(X)$, the


File Edit View Go Bookmarks Options Directory Window Help
Back Forward Home Edit Reload Images Open Print Find Stop
What's New? What's Cool? Destinations Net Search People Software
Load Image
Iterate Image

Netsite: http://www.informatik.uni-freiburg.de/~hartenst/Vis/Fraktal.html

Fractal Code Visualization

Move mouse on image to see rangeblocks (red) and matching domainblocks (blue). Click to select.

Selected Rangeblock

Matching Domainblock

Block (55/6) in Rangeblock 23 — Iso: 4 — Scale: 0.94 — Offset: 68.27

■ Use Scale-/Offset-Values

now shown: Original Image

Copyright (C) 1998 H. Hartenstein und R. Herz, Institut fuer Informatik, Universitaet Freiburg

Applet Fraktal running

Fig. 2. A screenshot of the *Fractal Code Visualizer* (a Java applet) that is available from `http://www.informatik.uni-leipzig.de/cgip/`. As input the visualizer takes an original image and a fractal code thereof. By moving the mouse pointer over the image, the borders of the range to which one is pointing are drawn as well as the borders of the corresponding domain. To the right the selected range and domain are depicted; here, the (reflected) domain is viewed with the corresponding luminance transformation ϕ applied.

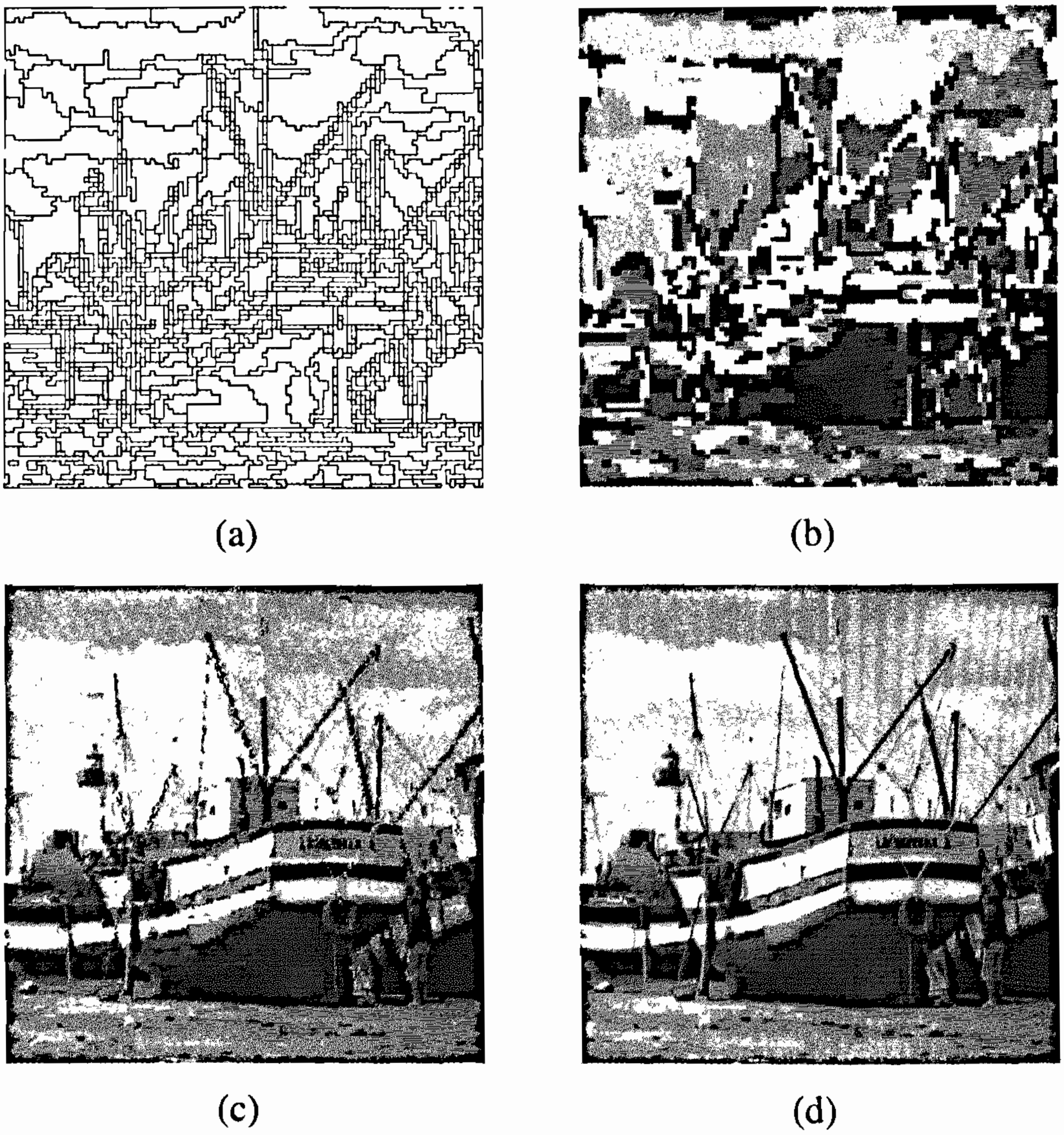


Fig. 3. Decoding of a fractal code for the standard test image 'Boat'. (a) Range partition, (b) first iteration, i.e., the operator applied to an all-black image, (c) third iteration, (d) 10th iteration.

space of square integrable functions on X with the usual metric, and set the parameter space to

$$\mathcal{P}^0 = \{p = ((s_1, o_1), \ldots, (s_{n_\mathcal{R}}, o_{n_\mathcal{R}}))| s_i, o_i \in \mathbb{R}, 1 \leq i \leq n_\mathcal{R}, \\ \text{s.t. } T_p \text{ is contractive in } \mathcal{L}^2(X)\}.$$

We first establish the existence of these derivatives and show that they are attractor functions of "vector fractal transform" operators (in the sense of hierarchical IFS [25, Chapter 5]). A knowledge of these derivatives permits the computation of the gradient vector of the error function $d_{\mathcal{F}}(f, \bar{f}_p)$ which, in turn, allows the use of gradient descent algorithms.

For the second part, we assume that one is given a target function and a range partition but no range-domain assignments. We now view the problem of optimal fractal coding as a combinatorial optimization problem, i.e., we model the space of feasible fractal codes via

$$\mathcal{P}^1 = \{p = ((z_1, s_1, o_1), \ldots, (z_{n_\mathcal{R}}, s_{n_\mathcal{R}}, o_{n_\mathcal{R}})) | 1 \leq z_i \leq n_D, \\ s_i \in Q(s), o_i \in Q(o), 1 \leq i \leq n_\mathcal{R}\}.$$

Here, each z_i represents an *address* for a domain block and $n_\mathcal{D}$ gives the (finite) number of domain choices per range. The sets $Q(s)$ and $Q(o)$ represent finite sets of feasible values for the scaling parameters and offsets, respectively. For practical applications one can assume that one is acting on a function space that is a finite-dimensional vector space. Thus, in order to guarantee convergence of the sequence of iterates $T^j f$, the constraint $|s_k| < 1, 1 \leq k \leq n_\mathcal{R}$ can be employed. We will show that the problem of determining in the parameter space a fractal code whose corresponding fractal transform operator gives the minimal attractor error is NP-hard, and, therefore, it cannot be solved in a practical amount of time.

3 Direct Attractor Optimization Based on Gradient Methods

3.1 Partial Derivatives of IFSM Attractor Functions with Respect to Luminance Parameters

Let us assume that the range partition and the range-domain assignments are given, i.e., the IFS maps $w_i, 1 \leq i \leq n_\mathcal{R}$, are fixed. Thus, the corresponding fractal transform operators T that are contractive in the space $\mathcal{F}(X) = \mathcal{L}^2(X)$ are parameterized via $\mathcal{P}^0$. We now consider the corresponding attractor functions $\bar{f}_p$ as functions not only of position but also of the luminance parameters p, i.e., we will write $\bar{f}(x, p)$ instead of $\bar{f}_p$.

Then, from Eq. (1) and with $p = (s_1, o_1, \ldots, s_{n_\mathcal{R}}, o_{n_\mathcal{R}}) \in \mathcal{P}^0$,

$$\bar{f}(x, p) \quad = \quad s_k \bar{f}(w_k^{-1}(x), p) + o_k, \quad x \in R_k. \tag{3}$$

Proposition 1. *The attractor $\bar{f}$ is continuous w. r. t. the fractal parameters s_l, o_l, $l = 1, 2, \ldots, n_\mathcal{R}$.*

The continuity of IFSM attractors with respect to grey level maps ϕ_l was proved in [12], using the methods described in [6]. It is straightforward to establish the continuity in terms of the luminance parameters s_l and o_l.

Proposition 2. *The set $\mathcal{P}^0$ is open.*

Proof. We prove that $\overline{\mathcal{P}^0} = \mathbb{R}^{2n_\mathcal{R}} - \mathcal{P}^0$ is closed. Let $p^{(n)} \in \overline{\mathcal{P}}$, $n = 1, 2, \ldots$, be a convergent sequence (in the topology of $\mathbb{R}^{2n_\mathcal{R}}$) with limit p. Each (infeasible) fractal code vector $p^{(n)} \in \overline{\mathcal{P}^0}$ defines a noncontractive fractal transform operator $T^{(n)} : \mathcal{L}^2(X) \to \mathcal{L}^2(X)$ with associated factor (cf. Eq. (2)) $C^{(n)} = \sum_{k=1}^{n_\mathcal{R}} c_k |s_k^{(n)}|$. Now, for each operator $T^{(n)}$, define its "optimal" Lipschitz factor as follows,

$$L^{(n)} = \sup_{y_1 \neq y_2} \frac{\| T^{(n)} y_1 - T^{(n)} y_2 \|_2}{\| y_1 - y_2 \|_2}. \tag{4}$$

From this definition and the noncontractivity of the $T^{(n)}$, it follows that $1 \leq L^{(n)} \leq C^{(n)}$ for all n. From the convergence of the code vectors $p^{(n)}$, it also follows that $\lim_{n \to \infty} C^{(n)} = C \geq 1$. So, from Proposition 1, the fractal transform T defined by the limit code vector p has associated factor C and Lipschitz factor $L \geq 1$. Therefore T is not contractive, implying that $p \notin \mathcal{P}^0$. Thus $\overline{\mathcal{P}^0}$ is closed, proving the proposition. □

Theorem 3. *The partial derivatives of the attractor $\bar{f}$ with respect to the fractal parameters s_l, o_l, $l = 1, 2, \ldots, n_\mathcal{R}$, exist at any point $p \in \mathcal{P}^0$.*

In the proof we need a special type of IFSM/fractal transform that involves "condensation" [21]. For a function $f \in \mathcal{F}(X)$, define Tf as follows: For all $x \in R_k$, $k = 1, 2, \ldots, n_\mathcal{R}$,

$$(Tf)(x) \;=\; s_k f(w_k^{-1}(x)) + \theta_k(x). \tag{5}$$

The functions $\theta_k(x)$ are known as *condensation* functions. Note that condensation functions do *not* affect the contractivity of T. The following result, which establishes the continuity of attractor functions with respect to condensation functions, is a simple consequence of Proposition 1.

Proposition 4. *Let T_1 and T_2 be contractive $n_\mathcal{R}$-map IFSM operators as in Eq. 5, with condensation functions $\theta^{(1)}(x)$ and $\theta^{(2)}(x)$, respectively, and identical scaling parameters s_k. Let $\bar{f}^{(1)}$ and $\bar{f}^{(2)}$, respectively, denote the fixed points of these operators. Then given an $\epsilon > 0$, there exists a $\delta > 0$ such that $\| \theta^{(1)} - \theta^{(2)} \|_2 < \delta$ implies that $\| \bar{f}^{(1)} - \bar{f}^{(2)} \|_2 < \epsilon$.*

Proof of Theorem 3: For any $p \in \mathcal{P}^0$, the associated fractal transform T is contractive. This implies that for any $f^{(0)} \in \mathcal{L}^2$, the sequence of functions defined by $f^{(n+1)} = Tf^{(n)}$ converges to $\bar{f}$, that is, $\| f^{(n)} - \bar{f} \|_2 \to 0$ as $n \to \infty$. Let $f^{(0)} = \theta$, where

$$\theta(x) = \sum_{k=1}^{n_\mathcal{R}} o_k I_{R_k}(x) \tag{6}$$

and $I_S(x)$ is the characteristic function of a subset $S \subset X$. Then, for $M \geq 0$, $f^{(M)} = T^{\circ M} f^{(0)}$ is given by

$$f^{(M)}(x, p) = \theta(x) + \sum_{n=1}^{M} \sum_{i_1, \ldots, i_n = 1}^{n_\mathcal{R}} s_{i_1} \cdots s_{i_n} \theta(w_{i_n}^{-1} \circ \cdots \circ w_{i_1}^{-1}(x)), \tag{7}$$

with the standard convention that $\theta(w_{i_n}^{-1} \circ \cdots \circ w_{i_1}^{-1}(x))$ equals zero if $w_{i_n}^{-1} \circ \cdots \circ w_{i_1}^{-1}(x)$ does not exist. The $f^{(M)}$ are partial sums of an infinite series that converge, in the $\mathcal{L}^2$ metric, to $\bar{f}$. Thus we can write

$$\bar{f}(x,p) = \theta(x) + \sum_{n=1}^{\infty} \sum_{i_1,\ldots,i_n=1}^{n_{\mathcal{R}}} s_{i_1} \cdots s_{i_n} \theta(w_{i_n}^{-1} \circ \cdots w_{i_1}^{-1}(x)), \qquad (8)$$

where the equation is understood in the $\mathcal{L}^2$ sense.

Now consider an $x \in R_k$ for some $k \in \{1,2,\ldots,n_{\mathcal{R}}\}$. Then the index i_1 in Eq. (7) must equal k (in order for $w_{i_1}^{-1}(x)$ to be defined). Therefore, Eq. (7) becomes

$$f^{(M)}(x,p) = \theta(x) + s_k f^{(M-1)}(w_k^{-1}(x),p), \qquad x \in R_k. \qquad (9)$$

For a given $l \in \{1,2,\ldots,n_{\mathcal{R}}\}$, we partially differentiate the terms in this equation with respect to s_l:

$$\frac{\partial f^{(M)}}{\partial s_l}(x,p) = s_k \left[\frac{\partial f^{(M-1)}}{\partial s_l}(w_k^{-1}(x),p)\right] + [f^{(M-1)}(w_k^{-1}(x),p)]\delta_{kl}. \quad (10)$$

Define the following $n_{\mathcal{R}}$-map IFSM operator T_l with condensation:

$$(T_l f)(x) \;=\; s_k f(w_k^{-1}(x)) + \xi_k(x), \qquad x \in R_k, \quad 1 \le k \le n_{\mathcal{R}}, \qquad (11)$$

where $\xi_k(x) = [\bar{f}(w_k^{-1}(x))]\delta_{kl}$ with $\delta_{kl} = 1$ if $k = l$ and zero otherwise. Since T is contractive, it follows that T_l is contractive in $\mathcal{L}^2$. (T and T_l have identical IFS maps and fractal parameters s_k.) Let $\bar{v}_l$ denote the fixed point of T_l. From Propositions 1 and 2, $\bar{v}_l$ is continuous with respect to the parameters s_k, in particular, s_l. We now show that $\bar{v}_l = \partial \bar{f}/\partial s_l$. (In what follows, for simplicity of notation, only x and s_l will be written explicitly in the list of independent variables.)

Note that Eq. (10) does not correspond to a single IFSM operator with condensation. However, since the functions $f^{(M)}$ converge to $\bar{f}$, it follows, from Proposition 4, that the sequence of functions $\partial f^{(M)}/\partial s_l$ converges to $\bar{v}_l$. That is, for a given $p \in \mathcal{P}^0$ and $\epsilon_1 > 0$, there exists an $M_1 > 0$ such that

$$\left\| \frac{\partial f^{(M)}}{\partial s_l}(x,s_l) - \bar{v}(x,s_l) \right\|_2 < \epsilon_1, \qquad \forall M > M_1. \qquad (12)$$

It is convenient to denote our reference point as

$$p^0 = (s_1^0,\ldots,s_{n_{\mathcal{R}}}^0, o_1^0,\ldots,o_{n_{\mathcal{R}}}^0) \in \mathcal{P}^0.$$

Let $N_l(\delta)$, $\delta > 0$, be a restricted neighborhood of the point p^0 in which only the element s_l is allowed to vary, i.e., $s_l \in I_\delta = [s_l^0 - \delta, s_l^0 + \delta]$, such that the corresponding vectors p lie in $\mathcal{P}^0$. (The existence of such a neighborhood is guaranteed since $\mathcal{P}^0$ is open.) Let $h \in \mathbb{R}$, with $|h| < \delta$. Then for each $x \in X$

there exists, by the Mean Value Theorem, a $\gamma^{(M)} \in I_h = [s_l^0 - h, s_l^0 + h]$, such that

$$f^{(M)}(x, s_l^0 + h) - f^{(M)}(x, s_l^0) = \frac{\partial f^{(M)}}{\partial s_l}(x, \gamma^{(M)})h. \tag{13}$$

Therefore,

$$\begin{aligned} \| f^{(M)}(x, s_l^0 + h) &- f^{(M)}(x, s_l^0) - h\bar{v}(x, s_l^0) \|_2 \\ &= h \left\| \frac{\partial f^{(M)}}{\partial s_l}(x, \gamma^{(M)}) - \bar{v}(x, s_l^0) \right\|_2 \\ &\leq h \left\| \frac{\partial f^{(M)}}{\partial s_l}(x, \gamma^{(M)}) - \bar{v}(x, \gamma^{(M)}) \right\|_2 \\ &\qquad + h \| \bar{v}(x, \gamma^{(M)}) - \bar{v}(x, s_l^0) \|_2 \\ &\leq h \left\| \frac{\partial f^{(M)}}{\partial s_l}(x, \gamma^{(M)}) - \bar{v}(x, \gamma^{(M)}) \right\|_2 \\ &\qquad + \max_{s_l \in I_h} h \| \bar{v}(x, s_l) - \bar{v}(x, s_l^0) \|_2 \, . \end{aligned} \tag{14}$$

Since I_δ is closed, there exists an $\overline{M} > 0$ such that the inequality in (12) is satisfied for all $M > \overline{M}$ at all $p \in N_l(\delta)$. Therefore, for a fixed $h \in (-\delta, \delta)$, we may take the limit $M \to \infty$ on both sides of (14) to yield

$$\left\| \frac{\bar{f}(x, s_l^0 + h) - \bar{f}(x, s_l^0)}{h} - \bar{v}(x, s_l^0) \right\|_2 \leq \max_{s_l \in I_h} \| \bar{v}(x, s_l) - \bar{v}(x, s_l^0) \|_2 \, . \tag{15}$$

Since $\bar{v}$ is continuous with respect to s_l, the right side term may be made arbitrarily small by choosing h sufficiently small, thus establishing the differentiability of $\bar{f}$ with respect to s_l at p^0.

The differentiability of $\bar{f}$ with respect to the o_l may be derived in a similar fashion. □

Remark: From Eq. (10) (and its analogue for differentiation with respect to o_l), the partial derivatives of $\bar{f}$ with respect to the fractal parameters s_l and o_l may be obtained by formally differentiating both sides of Eq. (3). For a fixed $x \in R_k$:

$$\frac{\partial \bar{f}}{\partial s_l}(x, p) = s_k \left[\frac{\partial \bar{f}}{\partial s_l}(w_k^{-1}(x), p) \right] + [\bar{f}(w_k^{-1}(x), p)]\delta_{kl}, \tag{16}$$

$$\frac{\partial \bar{f}}{\partial o_l}(x, p) = s_k \left[\frac{\partial \bar{f}}{o_l}(w_k^{-1}(x), p) \right] + \delta_{kl}. \tag{17}$$

Eqs. (3), (16) and (17) may be considered to define a $(2n_{\mathcal{R}} + 1)$-component "vector IFSM with condensation" that may be written in the following compact form:

$$\bar{\mathbf{f}} = \mathbf{T}\bar{\mathbf{f}}, \tag{18}$$

where

$$\bar{\mathbf{f}}(\mathbf{x},\mathbf{p}) = \left[\bar{f}(x,p), \frac{\partial \bar{f}}{\partial p_1}(x,p), \ldots, \frac{\partial \bar{f}}{\partial p_{2n_{\mathcal{R}}}}(x,p)\right]^t . \tag{19}$$

Now define the space $\mathcal{F}^{2n_{\mathcal{R}}+1}(X) = \{\mathbf{f} = (f_1, f_2, \ldots, f_{2n_{\mathcal{R}}+1}) \mid f_j \in \mathcal{F}(X)\}$ with associated metric $d_{\mathcal{F}^{2n_{\mathcal{R}}+1}}(\mathbf{f},\mathbf{g}) = \max_{1\le j\le 2n_{\mathcal{R}}+1} d_{\mathcal{F}}(f_j, g_j)$. Then $\mathbf{T} : \mathcal{F}^{2n_{\mathcal{R}}+1}(X) \to \mathcal{F}^{2n_{\mathcal{R}}+1}(X)$. For an $f \in \mathcal{F}^{2n_{\mathcal{R}}+1}(X)$,

$$(\mathbf{Tf})(x) \;=\; s_k \mathbf{f}(w_k^{-1}(x)) \;+\; \mathbf{e}_k \cdot \mathbf{f}(w_i^{-1}(x)) \;+\; \Theta_k(x), \quad x \in R_k. \tag{20}$$

The vector $[\mathbf{e}_k]^t = (0, 0, \ldots, 1, \ldots, 0)$, where the "1" occurs in the $(k+1)$st entry, represents the only "mixing" of components of $\mathbf{f}$ under the action of $\mathbf{T}$. The function $\Theta_k(x)$ represents a condensation vector composed of constant functions: $[\Theta_k(x)]^t = (o_k, 0, 0, \ldots, 1, \ldots, 0)$, where the "1" occurs in the $(n_{\mathcal{R}} + 1 + k)$th entry.

Proposition 5. *Suppose that T is contractive in $(\mathcal{F}(X), d_{\mathcal{F}})$. Then $\mathbf{T}$ is contractive in $\mathcal{F}^{2n_{\mathcal{R}}+1}(X)$. Its fixed point $\bar{\mathbf{f}}$ is given by Eq. (19), where $\bar{f}$ is the fixed point of T, see Eq. (3).*

From Banach's Fixed Point Theorem, contractivity of T allows the computation of its fixed point function $\bar{f}$ by means of iteration. The above proposition implies that all partial derivatives $\partial \bar{f}/\partial p_l$ may also be computed by iteration: Begin with a "seed" $\mathbf{f}^{(0)} \in \mathcal{F}^{2n_{\mathcal{R}}+1}(X)$ and construct the sequence of vector functions $\mathbf{f}^{(n+1)} = \mathbf{Tf}^{(n)}$, $n \geq 0$. The calculations are very complex: Except in special cases, $\bar{f}$ and its partial derivatives will have to be computed for all $x \in X$. This will be discussed in more detail below.

3.2 Experimental Image Coding Results

Let $f \in \mathcal{L}^2(X)$ again denote the target function we seek to approximate. For a given fractal code $p \in \mathcal{P}^0$ we will consider the squared $\mathcal{L}^2$ error

$$E(p) = \|f - \bar{f}_p\|_2^2.$$

We now employ the attractor $\bar{f}_p$, in particular the attractor $\bar{f}_{p^c}$ where p^c again denotes a collage error optimal fractal code, as a starting point and vary the fractal code parameters p in an attempt to decrease the error function $E(p)$ as much as possible. This was also the strategy of Dudbridge and Fisher [9], who employed the Nelder-Mead simplex algorithm. In their scheme, the error function $E(p)$ is computed at strategic points.

A knowledge of the partial derivatives of $\bar{f}_p$ with respect to the fractal parameters p permits the computation of elements of the gradient vector of E:

$$\frac{\partial E}{\partial p_l}(p) \;=\; -2\left\| f - \bar{f}_p - \frac{\partial \bar{f}_p}{\partial p_l}\right\|_2^2, \quad l = 1, 2, \ldots, 2n_{\mathcal{R}}. \tag{21}$$

This allows us to employ gradient-descent and related methods to search for local minima.

Practically speaking, however, the partial derivatives $\partial\bar{f}/\partial p_l(x,p)$ must be computed at all points (pixels) $x \in X$. In addition to an $n \times n$ matrix required to store an image, an additional $2n_{\mathcal{R}}$ $n \times n$ matrices are needed, in general, to store the derivatives at all pixels. Borrowing from the terminology of quantum chemists, this "full configuration interaction" will compute the total rate of change of the attractor — hence the approximation error — with respect to changes in all fractal parameters p_l *for a fixed set of domain-range pair assignments.* When applying a gradient descent method to minimize the error function E less storage is required. It suffices to provide one additional $n \times n$ matrix to sequentially compute each component of the gradient $(\partial E/\partial p_1, \ldots, \partial E/\partial p_{2n_{\mathcal{R}}})$.

We apply our method to the fractal transform scheme examined by Dudbridge and Fisher [9], designed to minimize the interdependency of range blocks. The following four 512 × 512 pixel images (8 bpp), used in [9], were also used in this study: *Lena*, *Boat*, *Mandrill* and *Peppers.*[1] Each image was partitioned into 4 × 4 pixel range blocks, with four neighboring range blocks sharing a common 8 × 8 pixel domain block, namely the one that consists of the four ranges. Therefore, for each image, the inverse problem separates into 64^2 independent problems, each involving an 8×8 pixel image with four range blocks R_k, hence 8 fractal parameters (four scaling and four offset values).

As in [9], for each test image we first used collage coding to determine a fractal code p^c that minimizes the collage error. We then used this code as a starting point for a gradient-descent method. The NAG [23] subroutine E04DKF, which performs a quasi-Newton conjugate gradient minimization, was used. It was also desirable to compare these results with the non-gradient calculations of [9]. However, since some of our collage error results differed from those of [9], we have independently carried out attractor optimization using the Nelder-Mead simplex algorithm. The NAG subroutine E04CCF was used.

In all cases, the simplex and gradient methods yielded almost identical improvements. A comparison with [9] reveals some nonnegligible differences, not only in the collage errors but also in the improvements obtained by the simplex method. In all cases, we improved on the results of [9]. In both the simplex as well as the gradient algorithms, the results are quite sensitive to the settings of the tolerance/accuracy parameters as well as the maximum number of iterations (*maxiter*) allowed. Generally the best performance was obtained when the tolerance parameters for the simplex and gradient subroutines were set to 10^{-5} and 10^{-6}, respectively. The parameter *maxiter* was set to 2000, which is virtually infinity.

[1] These 512 × 512 images may be retrieved by anonymous ftp from the Waterloo Fractal Compression Project site `links.uwaterloo.ca` in the appropriate subdirectories located in `ftp/pub/BragZone/GreySet2`.

In Table 1 we present the *peak-signal-to-noise-ratio* (PSNR) values associated with collage coding and subsequent simplex and gradient optimized attractor coding, along with the improvements in PSNR. The numbers in brackets represent the CPU time required for each calculation. (We emphasize that these numbers are presented for the purpose of comparison, since the computer codes themselves are not optimized.)

Table 1. Results of (a) collage coding and attractor optimization using (b) simplex and (c) gradient methods, the latter two using collage coding as a starting point. All results are expressed in PSNR (dB). The final two columns list the improvement in PSNR achieved by the simplex method obtained in this study and Ref. [9], respectively.

	Collage	Attractor optimization		ΔPSNR	ΔPSNR [9]
	attractor	Simplex	Gradient		
Lena	29.25	29.87 (301)	29.87 (229)	0.62	0.35
Boat	26.66	27.42 (300)	27.42 (299)	0.56	0.41
Mandrill	21.52	22.11 (532)	22.08 (1500)	0.59	0.33
Peppers	29.34	30.02 (277)	29.94 (591)	0.68	0.33

In an attempt to understand how good the initial estimate provided by collage coding actually is, we have performed simplex and gradient optimization calculations for another set of initial conditions, namely, piecewise constant approximations to the images. In this case, all s_l are initially set to zero and the o_l are simply the mean values of the range block. (Of course, in more general problems than the one studied here, there would remain the problem of assigning a domain block to each range block.) In Table 2, we present the results of these calculations. The first column gives the error associated with the initial piecewise constant approximation. The next two columns list the PSNR values of the optimized attractors obtained from the simplex and gradient methods along with the CPU times. The final column gives the PSNR improvement yielded by the better of the two methods.

Table 2. Results of (a) piecewise constant approximation (PCA) and attractor optimization using (b) simplex and (c) gradient methods, the latter two using the PCA as a starting point. All results are expressed in PSNR (dB). The final column lists the improvement in PSNR achieved by the better of methods (b) and (c).

	PCA	Simplex	Gradient	ΔPSNR
Lena	26.93	29.73 (421)	29.74 (288)	2.81
Boat	25.08	27.30 (452)	27.32 (618)	2.24
Mandrill	20.85	22.00 (663)	21.97 (3333)	1.15
Peppers	25.97	29.76 (420)	29.56 (2888)	1.79

We observe that the simplex and gradient methods, using such suboptimal initial conditions, i.e., piecewise constant approximations, yield approximations that are almost as good as those found from collage attractors. The worst case is *Peppers*, for which a 0.26 dB difference is found. For the others, the discrepancy is on the order of 0.1 dB.

Results of the gradient descent algorithms applied to fractal image encodings based on quadtree partitions can be found in our paper [28]. In these quadtree experiments we used the conjugate gradient algorithm from [26]. The major computational burden is the computation of the gradients required in each step, which allowed us to do experiments only with images of size 256 × 256. The gain obtained by the gradient descent method varied between 0.16 and 0.25 dB PSNR. However, the necessary quantization destroyed a large part of these gains. Thus, the achievable gains for fractal coding with the quadtree method are negligible.

4 On the Computational Complexity of Optimal Fractal Coding

In this section we will analyze the inverse problem of fractal coding from the computational complexity point of view,[2] i.e., we will consider optimal fractal coding as a discrete optimization problem. Thus, the support X is now given by $\{1, \ldots, n\}$,[3] and the space of functions $\mathcal{F}(X)$ equals $\mathbb{R}^n$. Instead of directly defining fractal transform operators acting on functions $f \in \mathbb{R}^n$, we simply interpret a function $f \in \mathbb{R}^n$ as a function on $[0,1)$ that is constant on each interval of $\mathcal{I} = \{[\frac{j}{n}, \frac{j+1}{n}) | 0 \leq j < n\}$. We will make the following assumptions that will allow us to easily translate 'back and forth' between discrete and continuous settings:

- Each range is a (connected) union of elements of $\mathcal{I}$.
- The affine mappings w_k^{-1} are of the form $w_k^{-1}(x) = \left(2x + \frac{j}{n}\right) \bmod 1, j \in \mathbb{Z}$; thus, the contraction factor of the mappings $w_k, 1 \leq k \leq n_{\mathcal{R}}$, is fixed to 0.5, and each domain is a union of elements of $\mathcal{I}$.

A fractal transform operator whose action is defined for $x \in I \subset R_k$, $I \in \mathcal{I}$, by

$$(Tf)(x) = s_k \cdot n \int_I f(w_k^{-1}(u))du + o_k \tag{22}$$

will again output a function that is constant on each interval of $\mathcal{I}$. Thus, an operator satisfying the above conditions and (22) can be regarded as a fractal operator acting on $\mathbb{R}^n$. Its basic difference to the original definition is the averaging over neighboring samples. Therefore, we will not distinguish

[2] For an introduction into the topic of computational complexity see, e.g., [14,24].

[3] For simplicity we restrict ourselves to the one-dimensional case. It is straightforward to extend all results and discussions to higher dimensions.

between the above operator and a 'truly' discrete operator, and write Tf also for functions $f \in \mathbb{R}^n$. Using the translation mechanism between piecewise constant functions and discrete functions we can now use the terms range, domain etc. also for the discrete case.

For the analysis of the computational complexity of optimal fractal coding we assume for simplicity that one is given a function $f \in \mathbb{R}^{m \cdot n_{\mathcal{R}}}$ that is uniformly partitioned into $n_{\mathcal{R}}$ ranges with m components each, i.e., $R_i = \{im, \ldots, (i+1)m - 1\}, 1 \le i \le n_{\mathcal{R}}$. The domains are non-overlapping and have twice the size of the ranges, i.e., the domains are given by $D_j = \{j \cdot 2m, \ldots, (j+1) \cdot 2m - 1\}, 1 \le j \le n_{\mathcal{D}} = \lfloor \frac{n_{\mathcal{R}}}{2} \rfloor$. We require the scaling parameters to have an absolute value smaller than 1 in order to guarantee convergence in the decoding. Thus, the set of feasible fractal codes for function f is given by

$$\mathcal{P}^1_{n_{\mathcal{R}}} = \{p = ((z_1, s_1, o_1), \ldots, (z_{n_{\mathcal{R}}}, s_{n_{\mathcal{R}}}, o_{n_{\mathcal{R}}}))| 1 \le z_i \le n_D,$$
$$s_i \in Q(s), o_i \in Q(o), 1 \le i \le n_{\mathcal{R}}\},$$

where $Q(s)$ and $Q(o)$ are finite sets of real values, and $|s| < 1$ for $s \in Q(s)$. The number of fractal codes in $\mathcal{P}^1_{n_{\mathcal{R}}}$ with different range-domain assignments is $(n_{\mathcal{D}})^{n_{\mathcal{R}}} = \lfloor \frac{n_{\mathcal{R}}}{2} \rfloor^{n_{\mathcal{R}}}$, since for each range one of the $n_{\mathcal{D}}$ domains is chosen. Thus, the number of feasible fractal codes grows exponentially with the number of ranges.

A fractal code $p^* \in \mathcal{P}^1_{n_{\mathcal{R}}}$ is called an optimal fractal code for function f (uniformly partitioned into $n_{\mathcal{R}}$ ranges) if

$$\|f - \bar{f}_{p^*}\|_2^2 \le \min_{p \in \mathcal{P}^1_{n_{\mathcal{R}}}} \|f - \bar{f}_p\|_2^2,$$

where $\bar{f}_p$ denotes again the attractor corresponding to the fractal code p.

Let us now formally define FRACCODE as the decision problem associated with the problem of optimal fractal coding.

FRACCODE
INSTANCE: Function $f \in \mathbb{Z}^n$ uniformly partitioned into $n_{\mathcal{R}}$ ranges with m components each, quantization levels $Q(s), Q(o)$, positive number Δ.
QUESTION: Is there an element p in $\mathcal{P}^1_{n_{\mathcal{R}}}$ (as defined above) whose attractor $\bar{f}_p$ satisfies $\|f - \bar{f}_p\|_2^2 \le \Delta$?

We will now prove that FRACCODE represents an NP-hard problem, thus, optimal fractal coding is NP-hard. Particularly, we will show that solving the FRACCODE problem is at least as hard as solving an instance of (unweighted) MAXCUT, i.e., we will give a polynomial transformation from MAXCUT to FRACCODE. The MAXCUT (decision) problem is defined as follows:

MAXCUT
INSTANCE: Undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $n_{\mathcal{V}}$ vertices and $n_{\mathcal{E}}$ edges, positive integer k.

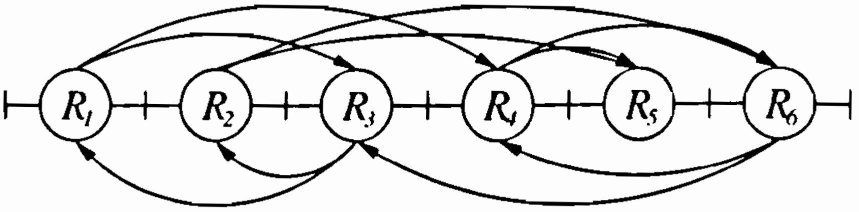


Fig. 4. Example of a dependency graph. Here, range R_5 is simply coded by an offset value.

QUESTION: Is there a partition of $\mathcal{V}$ into disjoint sets $\mathcal{V}_1$ and $\mathcal{V}_2$ such that the number of edges that have one endpoint in $\mathcal{V}_1$ and one endpoint in $\mathcal{V}_2$ is at least k?

Since MAXCUT is known to be NP-hard (to be precise, it is NP-complete, cf. [14, problem ND16] or [7, problem ND11]), it then follows that FRACCODE is also NP-hard.

Before proceeding to the formal proof, let us first explain intuitively what makes the problem of optimal fractal coding a hard problem.

The reconstruction quality for a function f on range R_i depends on how well the function is reconstructed on the domain for that range. Therefore, it depends on the reconstruction quality of the function on the ranges that are contained in the domain for range R_i, and so on. Those dependencies can be represented using a *dependency graph* as proposed in [8]. The dependency graph of a fractal code consists of the set of ranges $\{R_1, \ldots, R_{n_\mathcal{R}}\}$ as the set of vertices, and the set of edges is given by

$$\{(R_i, R_j) | R_j \text{ overlaps, fully or partially, with domain assigned to range } R_i\}.$$

An example is given in Figure 4. With the collage coding approach, each range is coded separately in a greedy fashion; the dependencies of the interference of the various range-domain maps are ignored by the collage coder which is the reason why collage coding is a suboptimal strategy. These dependencies are the reason why the determination of the optimal fractal code represents a computationally hard problem.

4.1 The Main Theorem

The reduction from MAXCUT will proceed as follows: given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $n_\mathcal{V}$ vertices and $n_\mathcal{E}$ edges, we will construct in polynomial time a signal $f(\mathcal{G}) \in \mathbb{R}^n$ with $n_\mathcal{R}$ ranges, sets $Q(s), Q(o)$ and a function $\Delta(\mathcal{G}, k)$ monotonically decreasing in k, $k \in \mathbb{N}$, such that the following holds:

Theorem 6. *$\mathcal{G}$ has a cut of size $\geq k \iff \exists p \in \mathcal{P}^1_{n_\mathcal{R}}$ such that $\|f(\mathcal{G}) - \bar{f}_p\|_2^2 \leq \Delta(\mathcal{G}, k)$.*

Thus, the question whether there exists a cut of a given cardinality k is reduced to the question whether there is an attractor $\bar{f}_p$ that approximates the signal $f(\mathcal{G})$ with an error of at most $\Delta(\mathcal{G},k)$. To prove Theorem 6 we will proceed in three steps. First, the construction of $f(\mathcal{G})$ and $\Delta(\mathcal{G},k)$ will be given. From the construction the $\Rightarrow$-direction will follow immediately:

Lemma 7. *$\mathcal{G}$ has a cut of size $\geq k \Rightarrow \exists p \in \mathcal{P}^1_{n_\mathcal{R}}$ such that $\|f(\mathcal{G}) - \bar{f}_p\|_2^2 = \Delta(\mathcal{G},k)$.*

As the last step we show the $\Leftarrow$-direction of Theorem 6 which is equivalent to the statement

$$\begin{aligned}&\mathcal{G} \text{ has a maximal cut of size smaller than } k\\ \Rightarrow\ &\nexists p \in \mathcal{P}^1_{n_\mathcal{R}} \text{ such that } \|f(\mathcal{G}) - \bar{f}_p\|_2^2 \leq \Delta(\mathcal{G},k).\end{aligned}$$

This in turn is equivalent to the following lemma:

Lemma 8. *$\mathcal{G}$ has a maximal cut of size $k \Rightarrow \nexists p \in \mathcal{P}^1_{n_\mathcal{R}}$ such that $\|f(\mathcal{G}) - \bar{f}_p\|_2^2 \leq \Delta(\mathcal{G},k+1)$.*

In Subsection 4.2 we give the construction of $f(\mathcal{G})$. The function Δ is given together with Lemma 7 in Subsection 4.3. Lemma 8 is shown in Subsection 4.4. Note that, for simplicity, in the following we will also call $f \restriction R_i$ a *range* and $f \restriction D_i$ a *domain.*

4.2 Construction of $f(\mathcal{G})$

In order to satisfy Theorem 6 we have to construct a signal $f(\mathcal{G})$ such that the approximation error resulting from the optimal attractor indicates whether or not the graph $\mathcal{G}$ has a cut of size at least k. The signal $f(\mathcal{G})$ will consist of five segments $S_0, \ldots, S_4$ that are designed as follows.

First of all, we assign to each vertex $\mathfrak{v} \in \mathcal{V}$ of the graph $\mathcal{G}$ a distinct signal, the *vertex ID.* IDs pertaining to different vertices will differ significantly from each other. The segments $S_1, \ldots, S_4$ are constructed as follows:

- Signal segment S_1 contains for each vertex $\mathfrak{v} \in \mathcal{V}$ four ranges as shown in Figure 5 a). The first and the third range contain the vertex ID for $\mathfrak{v}$, the second and the fourth range contain signals that are complementary to each other. The two complementary signals are used as binary *flags* and are denoted by B_1 and B_2.
- Segment S_2 contains two ranges for each vertex $\mathfrak{v}$ (cf. Figure 5 b)). The first half of the first range is again the vertex ID of $\mathfrak{v}$, shrunk to half its width. The rest of the two ranges equals zero.
- In the third segment, S_3, for each edge $(\mathfrak{v}_i, \mathfrak{v}_j) \in \mathcal{E}$ we have the following two ranges (cf. Figure 5 c)): The first quarter of the first range is the appropriately shrunk vertex ID of $\mathfrak{v}_i$, the first quarter of the second range contains the vertex ID of $\mathfrak{v}_j$. The rest of the two ranges is zero.

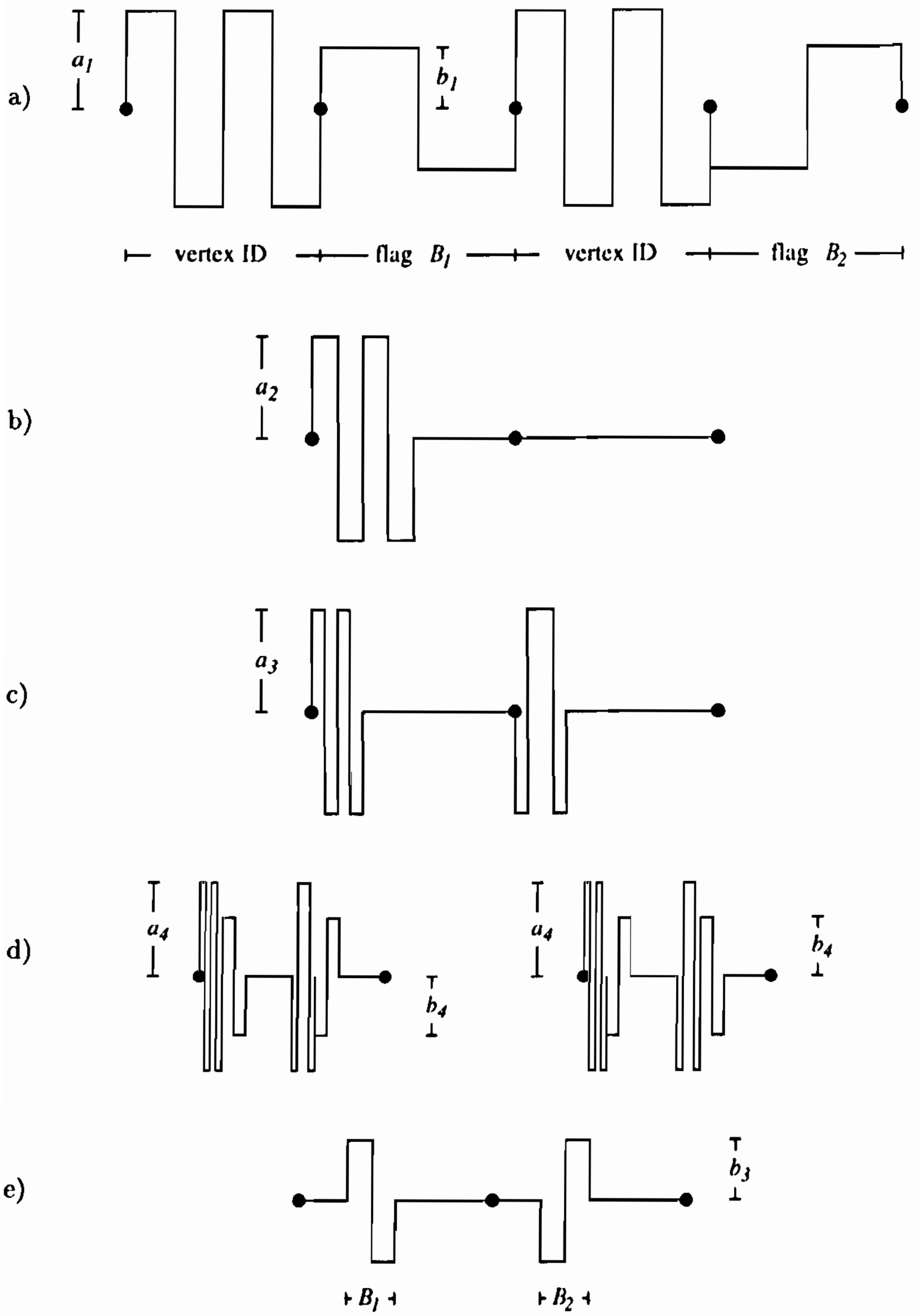


Fig. 5. Design of $f(\mathcal{G})$

- The fourth segment, S_4, contains two ranges for every edge in $(\mathfrak{v}_i, \mathfrak{v}_j) \in \mathcal{E}$ (cf. Figure 5 d)). Both ranges contain the vertex IDs of $\mathfrak{v}_i$ and $\mathfrak{v}_j$. Next to the vertex IDs copies of the flags are placed. In the first range, these are the flags B_1 and B_2, in the second range B_2 and B_1 (in this order).

The heights $a_1, \ldots, a_4$ of the signal are related by $a_2 = \sigma_1 \cdot a_1$, $a_3 = \sigma_2 \cdot a_2$, $a_4 = \sigma_3 \cdot a_3$. Furthermore, we set $b_i = \frac{a_i}{\sqrt{2}}$ for $i = 1, \ldots, 4$.[4] We set a_4 to some arbitrary, but fixed, constant. Thus, all parameters are completely determined by $a_4, \sigma_1, \sigma_2, \sigma_3$. The values of the σ_i will be determined in subsection 4.4. Note that due to this definition the signal does not necessarily consist of integer values. The assumption is that the parameters can be scaled by some sufficiently large factor and then rounded.

To motivate the construction, let us assume for the moment that the ranges of S_i have to be coded by domains from S_{i-1} for $i = 2, 3, 4$ and S_1 is given as side information. The vertex IDs will be designed in such a way that an ID mismatch will be *very* costly, i.e., when a fractal code assigns a domain to a range with a different vertex ID, this will result in a large reconstruction error for that range. Thus, for each range in S_2 the only 'possible' domains are the two domains with corresponding ID in S_1. Both contribute the same distortion in the attractor. Selecting one of them for each range corresponds to the partitioning of $\mathcal{V}$ into $\mathcal{V}_1$ and $\mathcal{V}_2$. The flag (B_1 or B_2, respectively) associated to the vertex $\mathfrak{v}$, therefore, indicates to which set of the partition $\mathfrak{v}$ belongs ($\mathcal{V}_1$ resp. $\mathcal{V}_2$). Again, each range of S_3 has to be coded by the domain of S_2 with the same vertex ID. In the attractor this third segment contains the information about which edges of the graph $\mathcal{G}$ belong to the cut. The segment S_4 will be used to *count* the number of these edges. An edge in the cut consists of a pair of vertices to which different flags (B_1 and B_2, or vice versa) have been assigned. In that case, we can find an exact match for one of the ranges in S_4 belonging to that edge. By doing so, the error of the attractor is coupled with the size of the cut.

In fact, the signal is hard to code since at segment S_2 it does not make any difference which of the two domains in S_1 with the same ID is chosen for each range, but the effect of the choice will affect the reconstruction error in segment S_4. The collage coder cannot decide which domain should be chosen because it does not take into account the implications of such a decision. Therefore, it simply uses some kind of tie breaking rule.

To make things explicit, we now give the remaining details of the construction of the signal $f(\mathcal{G})$. The IDs are built using the following lemma:

Lemma 9. *For each $\kappa \in \mathbb{N}$ there exists a binary code with κ codewords $\mathfrak{c}_1, \ldots, \mathfrak{c}_\kappa$, each of length $\ell = \mathcal{O}(\kappa)$, such that for $i \neq j$ the Hamming distances $d_{\mathcal{H}}(\mathfrak{c}_i, \mathfrak{c}_j)$ and $d_{\mathcal{H}}(\mathfrak{c}_i, \overline{\mathfrak{c}_j})$ equal $\ell/2$. $\overline{\mathfrak{c}_i}$ denotes the binary complement of $\mathfrak{c}_i$.*

[4] By following the proof backwards one can derive the feasible ratios between b_i and a_i; our choice facilitates calculations. More details are given in [17].

Proof. We will show by induction that the lemma holds for $\kappa = \ell = 2^q$, for all $q \in \mathbb{N}$. For all other κ simply choose κ of the codewords constructed for size $2^{\lceil \log_2 \kappa \rceil}$. To begin the induction, $\mathfrak{c}_1 = 0$ is such a code for $\kappa = 2^0$. For $\kappa = 2^{q+1}$ take the set $\{\mathfrak{c}_i\mathfrak{c}_i, \mathfrak{c}_i\overline{\mathfrak{c}_i} | 1 \leq i \leq 2^q\}$, where the $(\mathfrak{c}_i)_{1 \leq i \leq 2^q}$ form a code of the desired type of length 2^q. This gives a new binary code of size 2^{q+1} that is easily shown to have the desired property. □

Let $(\mathfrak{c}_i)_{1 \leq i \leq n_\mathcal{V}}$ be a binary code of $n_\mathcal{V}$ codewords of length ℓ constructed as in Lemma 9. From $(\mathfrak{c}_i)_{1 \leq i \leq n_\mathcal{V}}$ we build the binary code $\mathfrak{C}$ with codewords of length 2ℓ:

$$\mathfrak{C} := \{\mathfrak{c}_i\overline{\mathfrak{c}_i} | 1 \leq i \leq n_\mathcal{V}\}$$

$\mathfrak{C}$ has the property that two different codewords differ in half their bits and —as a consequence of Lemma 9— has the following features, which we will use in our calculations:

- Every codeword consists of ℓ 0s and ℓ 1s.
- For codewords $\mathfrak{c}_i\overline{\mathfrak{c}_i}, \mathfrak{c}_j\overline{\mathfrak{c}_j} \in \mathfrak{C}$ the following holds:
  - there are exactly $\frac{\ell}{2}$ positions where $\mathfrak{c}_i\overline{\mathfrak{c}_i}$ has a 0 and $\mathfrak{c}_j\overline{\mathfrak{c}_j}$ has a 1.
  - there are exactly $\frac{\ell}{2}$ positions where $\mathfrak{c}_i\overline{\mathfrak{c}_i}$ has a 1 and $\mathfrak{c}_j\overline{\mathfrak{c}_j}$ has a 0.

From the code $\mathfrak{C}$ we obtain the vertex IDs for segment S_1 as follows. Essentially, we interpret the 0s and 1s of the binary codewords as $-a_1$ and a_1. But in order to have unaliased geometrically shrunk versions for the vertex IDs in segments S_2, S_3, S_4, each value has to be repeated 8 times. Thus, the size m of a range has to be $m = 16\ell$. Therefore, the range size depends linearly on the number of vertices $n_\mathcal{V}$. We remark that the vertex IDs shown in Figures 5 and 7 are chosen for their simple shapes and are not constructed with the above approach. Also note that the vertex IDs have to be distinct from the binary flags. The above properties of the code $\mathfrak{C}$ guarantee that when for a range containing the vertex ID for $\mathfrak{v}_i$ a domain is assigned such that the vertex ID for $\mathfrak{v}_i$ is approximated by a vertex ID for $\mathfrak{v}_j, i \neq j$, this results in a large approximation error (cf. Figure 6). The proof in Subsection 4.4 depends heavily on this property.

In order to have all ingredients for coding segment S_1 without any distortion, we add a construction segment S_0 to the signal. For example, S_0 contains the signal parts that represent geometrically scaled copies of length $2m, 4m, \ldots, m^2$ of the ranges in S_1. Thus, we add construction segments $S_{0,0}, \ldots, S_{0,\log_2 m-1}$ where $S_{0,i}$ is built by repeating each component (sample) of S_1 $\frac{m}{2^i}$ times, $0 \leq i < \log_2 m$. We set $S_0 := S_{0,0} \ldots S_{0,\log_2 m-1}$. Clearly, the length of S_0 depends polynomially on the number of vertices $n_\mathcal{V}$.

For the edge counting to work we also need an extra block in S_3 of the shape sketched in Figure 5 e). Of course, this also leads to the addition of some construction blocks in segments S_0, S_1, S_2.

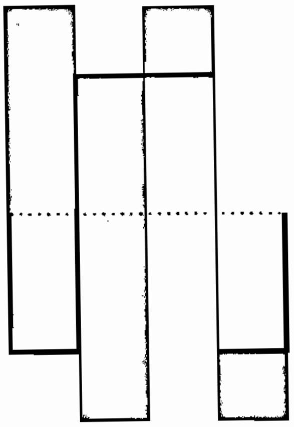


Fig. 6. Schematic representation of an ID mismatch; the grey-shaded area indicates the error.

4.3 Constructing an Attractor for $f(\mathcal{G})$

For the signal $f(\mathcal{G}) = S_0S_1S_2S_3S_4$ as described above we now give a transformation T_p that will later be shown to generate the optimal attractor. In the fractal code p a range in S_i is assigned a domain in $S_{i-1}, i = 1, \ldots, 4$. We will be able to determine easily the attractor of T_p, since there will be no need for iterating the operator T_p, i.e., the dependency graph corresponding to T_p will not contain any cycles.

First of all, the segments S_0, S_1 can be coded without any distortion. By hypothesis, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ has a cut of cardinality k by partitioning $\mathcal{V}$ into $\mathcal{V}_1$ and $\mathcal{V}_2$. For a range in S_2 containing a (geometrically shrunk) vertex ID we choose the domain in S_1 with the same ID and the flag set in accordance to the graph partition. The scaling parameter and offset are set to $s = \frac{2}{3}\sigma_1$ and $o = 0$. In this way the maximal height of the attractor on S_2 is $\frac{2}{3}$ of the maximal height of the original signal on S_2 (cf. Figure 7(a)). On the first half of the range an error of $\frac{m}{2}(a_2 - \frac{2}{3}a_2)^2 = \frac{m}{18}a_2^2$ occurs, on the other half of the range the error is $\frac{m}{2}(\frac{2}{3}\frac{a_2}{\sqrt{2}})^2 = \frac{m}{9}a_2^2$. Therefore, on each range of segment S_2 that contains a vertex ID an error of $\frac{m}{6}a_2^2$ occurs. Thus, the total distortion of Ω_F in segment S_2 is

$$n_{\mathcal{V}} \cdot \frac{m}{6}a_2^2.$$

For each range in S_3 we choose the corresponding domain of S_2, i.e., the one with the same vertex ID, and scale it using $s = \sigma_2, o = 0$. The error introduced in segment S_3 then is

$$2n_{\mathcal{E}} \cdot \frac{m}{12}a_3^2.$$

The distortion in segment S_4 depends on the size of the cut k. For each edge there are two ranges in S_4 differing only in the flags. Depending on whether or not an edge belongs to the cut, we proceed as follows:

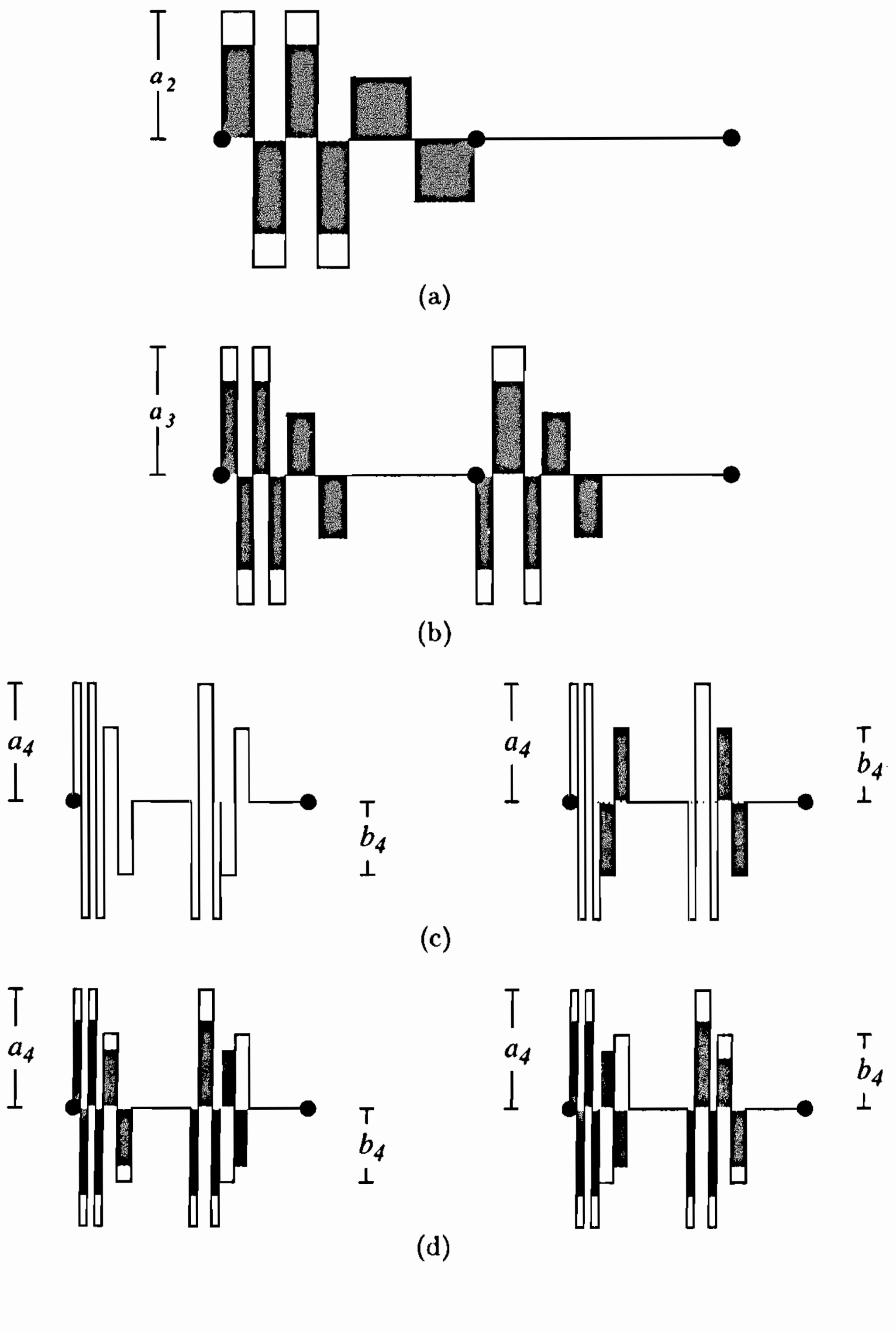


Fig. 7. Schematic representation of the attractor of F as defined in Section 4.3 (the attractor is given by the grey-shaded signal); (a) ranges of S_2, (b) ranges of S_3, (c) the two ranges in S_4 corresponding to an edge when the edge is in the cut, and (d) when the edge is not in the cut.

- *Edge belongs to cut.* In this case, one of the two ranges can be coded without any distortion by the corresponding domain in S_3. For this mapping the luminance parameters are $s = \frac{3}{2}\sigma_3$ and $o = 0$. The second range will be coded by the extra block as exemplified in Figure 5 e) $(s = \sigma_3, o = 0)$ yielding a distortion of $\frac{m}{4}a_4^2$.
- *Edge does not belong to cut.* In this case, we code both of the ranges in S_4 with the corresponding domain in S_3 $(s = \sigma_3, o = 0)$ obtaining a total error of

$$2m\left[\frac{1}{4}\left(a_4 - \frac{2}{3}a_4\right)^2 + \frac{1}{8}\left(b_4 - \frac{2}{3}b_4\right)^2 + \frac{1}{8}\left(b_4 + \frac{2}{3}b_4\right)^2\right] = m \cdot \frac{5}{12}a_4^2.$$

Therefore, the error introduced in S_4 by an edge that is not in the cut is $\frac{5m}{12}a_4^2 = \frac{m}{4}a_4^2 + \frac{m}{6}a_4^2 > \frac{m}{4}a_4^2$. Thus, the total error introduced in segment S_4 is $(\frac{1}{4}n_{\mathcal{E}} + \frac{1}{6}(n_{\mathcal{E}} - k))ma_4^2$. We define $\Delta(\mathcal{G}, k)$ as the distortion made by the attractor in all segments of the signal:

$$\Delta(\mathcal{G}, k) := m\left[\frac{1}{6}n_{\mathcal{V}}a_2^2 + \frac{1}{6}n_{\mathcal{E}}a_3^2 + (\frac{1}{4}n_{\mathcal{E}} + \frac{1}{6}(n_{\mathcal{E}} - k))a_4^2\right].$$

With this definition we (trivially) obtain Lemma 7. Now, using this definition we have to show the correctness of Lemma 8.

4.4 Proof of Lemma 8

We have to show that when $\mathcal{G}$ has a maximal cut of size k then no fractal code in $\mathcal{P}^1_{n_\mathcal{R}}$ leads to a distortion smaller or equal to $\Delta(\mathcal{G}, k+1)$. Let assume, on the contrary, that the graph $\mathcal{G}$ has a maximal cut of size k, but there exists a fractal code $p' \in \mathcal{P}^1_{n_\mathcal{R}}$ such that the attractor $\bar{f}_{p'}$ satisfies $\|f(\mathcal{G}) - \bar{f}_{p'}\|_2^2 \leq \Delta(\mathcal{G}, k+1)$. From Lemma 7 we know that there is an attractor $\bar{f}$ with $\|f(\mathcal{G}) - \bar{f}\|_2^2 = \Delta(\mathcal{G}, k)$. Obviously, $\bar{f}_{p'}$ is closer to the original signal $f(\mathcal{G})$ than $\bar{f}$. Consequently it must approximate $f(\mathcal{G})$ better on at least one of the segments of the signal, S_0, S_1, S_2, S_3, S_4. By setting $\sigma_1, \sigma_2, \sigma_3$ — depending only on the input graph $\mathcal{G}$— we will enforce that $\bar{f}_{p'}$ cannot be a better approximation than $\bar{f}$ on any part of the signal. Thus, our hypothesis is false and the lemma is proven.

Let us first assume that the ranges of S_2 have to be coded by domains from S_1, ranges from S_3 by domains from S_2 and ranges from S_4 by domains from S_3. At the end of the proof we will indicate how to remove this restriction.

Case 1: *$\bar{f}_{p'}$ is better than $\bar{f}$ on S_0 or S_1*
Since the difference of $\bar{f}$ and $f(\mathcal{G})$ is zero on S_0 and S_1, no improvement is possible, and, therefore, case 1 cannot occur.

Case 2: *$\bar{f}_{p'}$ is better than $\bar{f}$ on S_2*
For simplicity, first assume that $\bar{f}_{p'}$ is identical to $f(\mathcal{G})$ on part S_1. For a range R of segment S_2 that contains a vertex ID there are two possibilities for choosing a domain D:

1. When selecting a domain with a fitting vertex ID, the incurred error is (depending on scaling factor s and offset o)

$$\begin{aligned} E_{D,R}(s,o) = \frac{m}{4}\Big((a_2-(a_1\cdot s+o))^2+(-a_2-(-a_1\cdot s+o))^2+ \\ (b_1\cdot s+o)^2+(-b_1\cdot s+o)^2\Big). \end{aligned}$$

   Solving this equation for the optimal values of s and o yields $\tilde{s}=\frac{2}{3}\sigma_1, \tilde{o}=0$. This leads to an error of $E_{D,R}(\frac{2}{3}\sigma_1,0)=\frac{m}{6}a_2^2$ for the range R.
2. When selecting a domain with an incorrect vertex ID, the error will be

$$\begin{aligned} E_{D,R}(s,o) = \frac{m}{8}\Big((a_2-(a_1\cdot s+o))^2+(-a_2-(-a_1\cdot s+o))^2+ \\ (a_2-(-a_1\cdot s+o))^2+(-a_2-(a_1\cdot s+o))^2+ \\ 2\cdot(b_1\cdot s+o)^2+2\cdot(-b_1\cdot s+o)^2\Big). \end{aligned}$$

   Again, solving for optimal s, o yields $\tilde{s}=\tilde{o}=0$ with an error of $E_{D,R}(0,0) = \frac{m}{2}a_2^2$, three times the error incurred when matching correct IDs. Here, we have used the properties of the binary code $\mathfrak{C}$ of Subsection 4.2.

Thus, the error of $\bar{f}_{p'}$ on S_2 is at least $(n_{\mathcal{V}}+2l)\cdot\frac{1}{6}a_2^2\cdot m$, where l is the number of incorrect ID assignments. We choose σ_2 so small that the error made by one ID mismatch is larger than the error made by $\bar{f}$ in segments S_3 and S_4:

$$\begin{aligned} 2\cdot\frac{1}{6}a_2^2 &> \frac{1}{12}n_{\mathcal{E}}a_3^2+(\frac{1}{4}n_{\mathcal{E}}+\frac{1}{6}(n_{\mathcal{E}}-0))a_4^2 \\ \iff \frac{1}{3}a_2^2 &> \frac{1}{6}n_{\mathcal{E}}\sigma_2^2a_2^2+\frac{5}{12}n_{\mathcal{E}}\sigma_2^2\sigma_3^2a_2^2 \\ \iff \frac{1}{3} &> \sigma_2^2\left(\frac{1}{6}+\frac{5}{12}\sigma_3^2\right)n_{\mathcal{E}} \\ \iff \frac{2}{\sqrt{(2+5\sigma_3^2)n_{\mathcal{E}}}} &> \sigma_2 \qquad (23) \end{aligned}$$

Therefore, l must equal zero, since otherwise the error incurred in segment S_2 alone would be larger than $\Delta(\mathcal{G},k)$.

Let us now deal with the assumption that $\bar{f}_{p'}$ equals $f(\mathcal{G})$ on segment S_1. Note that the difference between $\bar{f}_{p'}$ and $f(\mathcal{G})$ on S_1 has to be less than $\Delta(\mathcal{G},k)$, and this value does not depend on σ_1. By choosing σ_1 sufficiently small, we can assure that the error of $\Delta(\mathcal{G},k)$ is very small relative to a_1. This relative error will then change our calculations slightly. But by scaling σ_1 we can make these differences arbitrarily small, in fact, significantly smaller than $\Delta(\mathcal{G},k)-\Delta(\mathcal{G},k+1)$.

Case 3: *$\bar{f}_{p'}$ is better than $\bar{f}$ on S_3*

First, we can assume that $\bar{f}_{p'}$ looks essentially like $\bar{f}$ on S_2. This is because by choosing σ_2 small enough, any difference that is noticeable after scaling down a domain from S_2 would mean a large additional error in the domain, larger than any potential savings in S_3 and S_4. Thus, we can assume that $\bar{f}_{p'}$ is identical to $f(\mathcal{G})$ on part S_2. For a range R of segment S_3 there are two possibilities for choosing a domain D:

1. When selecting a domain with a fitting vertex ID, the incurred error is

$$E_{D,R}(s,o) = \frac{m}{8}\Big((a_3 - (\frac{2}{3}a_2 \cdot s + o))^2 + (-a_3 - (-\frac{2}{3}a_2 \cdot s + o))^2 + (\frac{2}{3}b_2 \cdot s + o)^2 + (-\frac{2}{3}b_2 \cdot s + o)^2\Big).$$

   The factor $\frac{2}{3}$ comes into play, since we compare the range against the *reconstructed* domain, i.e., against the attractor $\bar{f}_{p'}$ on S_2. Solving this equation for the optimal values of s and o yields $\tilde{s} = \sigma_1, \tilde{o} = 0$. This leads to an error of $E_{D,R}(\sigma_1, 0) = \frac{m}{12}a_3^2$ for the range R.
2. When selecting a domain with an incorrect vertex ID, the error will be

$$E_{D,R}(s,o) = \frac{m}{16}\Big((a_3 - (\frac{2}{3}a_2 \cdot s + o))^2 + (-a_3 - (-\frac{2}{3}a_2 \cdot s + o))^2 + (a_3 - (-\frac{2}{3}a_2 \cdot s + o))^2 + (-a_3 - (\frac{2}{3}a_2 \cdot s + o))^2 + 2 \cdot (\frac{2}{3}b_2 \cdot s + o)^2 + 2 \cdot (-\frac{2}{3}b_2 \cdot s + o)^2\Big).$$

   Solving for optimal s and o yields $\tilde{s} = \tilde{o} = 0$ with an error of $E_{D,R}(0,0) = \frac{m}{4}a_3^2$.

The error of $\bar{f}_{p'}$ on S_3 is at least $(2n_\mathcal{E} + 2l) \cdot \frac{m}{12}a_3^2$, where l is the number of incorrect ID assignments. Again, by choosing σ_3 small enough, we can assure that if we use incorrect IDs, the error will be larger than the error made by $\bar{f}$ in S_4 (which does not depend on σ_3); thus, the total error would be larger than the error of $\bar{f}$:

$$\begin{aligned} \frac{1}{6}a_3^2 &> \left(\frac{1}{4}n_\mathcal{E} + \frac{1}{6}n_\mathcal{E}\right) a_4^2 \\ \Longleftrightarrow \quad \frac{1}{6}a_3^2 &> \left(\frac{1}{4}n_\mathcal{E} + \frac{1}{6}n_\mathcal{E}\right) \sigma_3^2 a_3^2 \\ \Longleftrightarrow \quad \frac{1}{6} &> \frac{5}{12}n_\mathcal{E}\sigma_3^2 \\ \Longleftrightarrow \quad \sqrt{\frac{2}{5n_\mathcal{E}}} &> \sigma_3. \end{aligned}$$

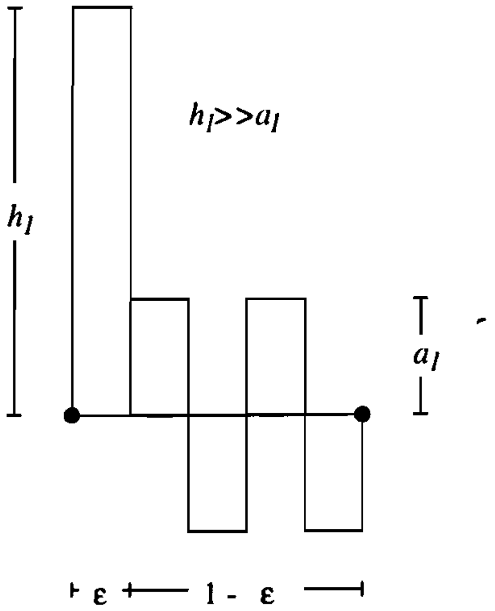


Fig. 8. Adding a peak to the ID

Case 4: *$\bar{f}_{p'}$ is better than $\bar{f}$ on S_4*
Again, we can assume that $\bar{f}_{p'}$ and $\bar{f}$ look 'the same' on segment S_3.

We will now examine what error the possible domain-range pairings will incur. To this end, we distinguish two cases: first, if an edge belongs to the cut, i.e., the flags of the two vertices are different, and second, if the flags are the same.

1. *Edge belongs to cut.* In this case one of the two edge copies in S_4 can be mapped with error zero. As for the other copy, by computing the optimal transformation parameters for all possible domains in S_3, we see that mapping the extra block on the range yields the minimum error of $\frac{1}{4}a_4^2 \cdot m$.
2. *Edge does not belong to cut.* In this case there exists no exact matching domain in S_3 for the two edge copies in S_4. Thus, these ranges can only be coded with a mismatched ID, wrong flag or the extra block. By computing the optimal transformation parameters for all possible domains we see that the error for each of the two edge copies is at least $\frac{5}{24}a_4^2 \cdot m$ for a total of $\frac{5}{12}a_4^2 \cdot m$ (we omit the details of the calculations) .

Thus, the error of $\bar{f}_{p'}$ in S_4 is at least $(n_{\mathcal{E}} \cdot \frac{1}{4}a_4^2 + (n_{\mathcal{E}} - k) \cdot \frac{1}{6}a_4^2)m$, which is exactly the error of $\bar{f}$.

It remains to be shown how one can assure that ranges from segment S_i are only coded by ranges from segment S_{i-1} for $i = 2, 3, 4$. This can be achieved by a slight modification of the signal $f(\mathcal{G})$. The basic idea is the following: We add at the left end of each ID in segment S_1 a peak of height h_1 and width ϵ as depicted in Figure 8 (the rest of the IDs are shrunk accordingly); we also add peaks of height h_i and width $\epsilon/2^{i-1}$ to the IDs of segments S_i for $i = 2, 3, 4$. Now, by appropriately choosing the heights $h_1, \ldots, h_4$ we can achieve that range-domain assignments other than those considered in our proofs above lead to arbitrarily large errors: when a range of $S_i, i \in \{1, 2, 3, 4\}$, containing an ID is assigned a domain from a segment other than S_{i-1}, the corresponding peak will differ with respect to the peak support in such a way that a large error occurs no matter what the rest of the domain and range looks like. Thus, we exploit the fact that the *geometric* scaling factor is fixed. At the same time, all argumentations of this and the previous subsection can be easily translated to the case of the 'peak-added' signal.

This consideration concludes our proof of Lemma 8. □

5 Further Results

In this section we survey further results related to optimal fractal compression and collage coding. We state that collage coding does not constitute an approximating algorithm for the problem of optimal fractal coding, derive a lower bound on the optimal attractor error, and devise an 'annealing' scheme for improved fractal coding.

5.1 Approximability

The NP-hardness result of the last section poses the question of whether the problem FRACCODE admits an approximating algorithm. In [27] we have shown that the method of collage coding does not represent an approximating algorithm, i.e., we have shown that there exits no (finite) constant $\rho > 0$ such that

$$\frac{d_{\mathcal{F}}(f, \bar{f}_{p^c})}{d_{\mathcal{F}}(f, \bar{f}_{p^*})} < \rho \quad \forall f \in \mathcal{F},$$

where again the attractor for a collage error optimal code is denoted by $\bar{f}_{p^c}$, and a truly optimal attractor is given by $\bar{f}_{p^*}$. In other words, it is possible that the ratio of the collage attractor error to the optimal attractor error can be arbitrarily large (for more details, cf. [17]). While this result shows that the collage coding strategy has fundamental shortcomings, it remains open whether *near-optimal* fractal coding is possible.

5.2 Anti-Collage-Theorem

Another interesting result concerning collage coding is given by the "Anti-Collage-Theorem" [28] that provides a lower bound for the collage attractor error in terms of the collage error. The generalized formulation is as follows:

Proposition 10. *Given (Y, d_Y) a complete metric space. Let $T : Y \rightarrow Y$ be Lipschitz, i.e., there exists an $L_T \geq 0$ such that $d_Y(Ty_1, Ty_2) \leq L_T d_Y(y_1, y_2)$ for all $y_1, y_2 \in Y$. As well, assume that $\bar{y}$ is a fixed point of T. Then for any $y \in Y$,*

$$d_Y(y, \bar{y}) \geq \frac{1}{1 + L_T} d_Y(y, Ty). \tag{24}$$

Proof. From the triangle inequality:

$$\begin{aligned} d_Y(y, Ty) &\leq d_Y(y, \bar{y}) + d_Y(\bar{y}, Ty) \\ &\leq d_Y(y, \bar{y}) + L_T(\bar{y}, y), \end{aligned} \tag{25}$$

from which the desired result follows.

Thus, for nonzero collage error, there is no chance that the error can be small "by accident."

5.3 Annealing

The result of Section 4 has shown that optimal fractal coding represents an intractable problem. However, the results obtained with the gradient-based direct attractor optimization (Section 3) indicate that improvements over collage coding are feasible. But with this approach only the luminance parameters are modified. As a next step one would like to include the domain addresses in the optimization as well. Unfortunately, even when the domain addresses would be considered as continuously varying entities, they cannot be included into the gradient-based optimization method for two reasons: the complexity would be prohibitive and the chance to get trapped in a local optimum would also be very high. Thus, in order to include the domain addresses in the optimization, one has to deal with the discrete problem.

In [15] a method is proposed for *local iterative improvements* of a fractal code. The basic idea is as follows: one tries to improve a given fractal code by selecting a single range and modifying the corresponding domain address and luminance parameters. The new 'candidate code' is then decoded, and one checks whether an improvement over the original code has occurred. When this is the case, the 'candidate code' is used as the new original code and the procedure is repeated with another range. The way the fractal code for a range is modified follows the proposal of Barthel [5] and Lu [20]: the domain search is performed in the attractor of the original code (not in the image to be coded as it is the case for the collage coding strategy).

For a practical application computational efficiency is crucial since the naive straightforward implementation leads to unacceptably long compute times. Here are some of the issues that are addressed in [16]:

- In what order should the range blocks be processed?
- Can one restrict the search for a matching domain to a few candidate blocks which are determined a priori in a preprocessing step?

- The decoder, which is an integral component of the coder, must be accelerated by exploiting the fact that the code changes only locally during one iteration of the algorithm.

The results (cf. Table 3) show significant improvements of about 0.6 dB over standard collage coding.

Table 3. Results for quadtree encodings of the 512 × 512 Boat image using the annealing procedure. Shown are the number of ranges of the partition, the compression ratio, the attractor error obtained using the collage coding strategy, the attractor error obtained with the additional annealing procedure, and the observed gain in PSNR.

No. of ranges	Comp. ratio	Collage coding (dB PSNR)	Annealing (dB PSNR)	Δ PSNR
4510	17:1	31.94	32.54	0.60
3352	23:1	30.19	30.77	0.58
2560	30:1	28.89	29.51	0.62
1972	38:1	27.83	28.51	0.68

6 Summary

In this paper we have reported on approaches to the inverse problem of fractal compression from two different directions using two different mathematical methodologies. In the first part we have derived the theoretical foundations necessary for any application of differentiable methods for attractor error reduction in fractal compression, namely

- the establishment of the differentiability of the attractor as a function of its (real valued) scaling and offset parameters, and
- the feasibility of gradient computation by iteration of a properly defined vector Iterated Function System with gray level Maps.

Moreover, we have implemented gradient descent algorithms for the problem and reported computational results for a few test cases. While the computer programs have demonstrated that the methods work in practice, the outcomes, however, are not promising. Although gains for the encoding based on the method of Dudbridge and Monro are around one half of a dB in PSNR, the conceptually less complex method using a simplex hill climbing algorithm performs just as well at the same cost in terms of computation time.

In the second part of the paper we have analyzed the computational complexity of the inverse problem, i.e., we have considered fractal compression as a discrete optimization problem and have analyzed the complexity of determining for a given function the fractal code —out of a class of feasible fractal codes— that achieves the least attractor error. Here, we have been able to prove that the problem is inherently intractable, i.e., NP-hard. This explains the predominant use of the suboptimal collage coding strategy, and, unfortunately, limits the prospects of improving fractal compression by searching for optimal fractal codes.

References

1. Barnsley, M. F., *Fractals Everywhere*, Academic Press, New York, 1988.
2. Barnsley, M. F., *Fractal modeling of real world images*, in: *The Science of Fractal Images*, H.-O. Peitgen and D. Saupe (eds.), Springer-Verlag, New York, 1988.
3. Barnsley, M.F., Ervin, V., Hardin, D., and Lancaster. J., *Solution of an inverse problem for fractals and other sets*, Proc. Nat. Acad. Sci. USA **83** (1985) 1975–1977.
4. Barnsley, M. F., Hurd, L., *Fractal Image Compression*, AK Peters, Wellesley, 1993.
5. Barthel, K. U., *Festbildcodierung bei niedrigen Bitraten unter Verwendung fraktaler Methoden im Orts- und Frequenzbereich* (Dissertation, Technische Universität Berlin), Wissenschaft & Technik Verlag, Berlin, 1996.
6. Centore, P., Vrscay, E.R., *Continuity of attractors and invariant measures for Iterated Function Systems*, Canad. Math. Bull. **37**(3) (1994) 315–329.
7. Crescenzi, P., Kann, V., *A compendium of NP optimization problems*, Technical report SI/RR-95/02, Universita di Roma "La Sapienza", 1995.
8. Domaszewicz, J., Vaishampayan, V. A., *Graph-theoretical analysis of the fractal transform*, Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, vol. 4, Detroit, 1995.
9. Dudbridge, F., Fisher, Y., *Attractor optimixation in fractal image encoding*, Proc. Third Fractals in Engineering Conference, Arcachon, France, May, 1997.
10. Fisher, Y., *Fractal Image Compression, Theory and Application*, Springer-Verlag, New York, 1995.
11. Fisher, Y. (ed.), *Fractal Image Coding and Analysis*, Proc. NATO ASI, Springer-Verlag, New York, 1998.
12. Forte, B., Vrscay, E.R., *Solving the inverse problem for function and image approximation using Iterated Function Systems*, Dyn. Cont. Disc. Imp. Syst. **1** (1995) 177–231.
13. Forte, B., Vrscay, E.R., *Inverse problem methods for generalized fractal transforms*, in *Fractal Image Encoding and Analysis*, ed. Y. Fisher, Springer Verlag, Heidelberg, 1998.
14. Garey, M. R., Johnson, D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York, 1979
15. Hamzaoui, R., Hartenstein, H., Saupe, D., *Local iterative improvement of fractal image codes*, Image & Vision Computing **18** (2000) 565–568.
16. Hamzaoui, R., Saupe, D., Hiller, M., *Fast code enhancement with local search for fractal image compression*, Proc. IEEE Int. Conf. on Image Processing, Vancouver, Sept. 2000.
17. Hartenstein, H., *Topics in Fractal Image Compression and Near-Lossless Image Coding*, Doctoral Dissertation, Institut für Informatik, Universität Freiburg, 1998.

18. Hutchinson, J., *Fractals and self-similarity*, Indiana Univ. Math. J. **30** (1981) 713–714.
19. Jacquin, A. E., *A Fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding*, Doctoral Dissertation, Georgia Institute of Technology, August 1989.
20. Lu, N., *Fractal Imaging*, Academic Press, New York, 1997.
21. Mendivil, F., Vrscay, E.R., *Correspondence between fractal-wavelet transforms and Iterated Function Systems with Grey-level Maps*, in *Fractals in Engineering*, ed. J. Lévy Véhel, E. Lutton and C. Tricot, Springer, New York, 1997.
22. Monro, D. M., Dudbridge, F., *Fractal block coding of images*, Electronics Letters **28**,11 (1992) 1053–1055.
23. NAG Fortran Library, The Numerical Algorithms Group Ltd, Oxford, UK.
24. Papadimitriou, C. H., *Computational Complexity*, Addison-Wesley, Reading, Mass., 1994
25. Peitgen, H.-O., Jürgens, H., Saupe, D., *Chaos and Fractals*, Springer-Verlag, New York, 1992.
26. Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P., *Numerical Recipes in C,* Second Edition, Cambridge University Press, 1992.
27. Ruhl, M., Hartenstein, H., *Optimal fractal coding is NP-hard*, in Proc. IEEE Data Compression Conference, ed. J. Storer and M. Cohn, Snowbird, Utah, 1997, 261–270.
28. Vrscay, E. R., Saupe, D., *Can one break the "collage barrier" in fractal image coding?*, in: *FRACTALS: Theory and Applications in Engineering*, Proceedings, Delft, June 1999, M. Dekking, J. Levy Vehel, E. Lutton, C. Tricot (eds.), p. 289–305, Springer-Verlag, Heidelberg, 1999.
29. Williams, R. F., *Compositions of contractions*, Bol. Soc. Brasil. Mat. **2** (1971) 55–59.
30. Withers, D., *Newton's method for fractal approximation*, Constructive Approximation **5** (1989) 151-170.