

# FRACTAL IMAGE COMPRESSION WITH FAST LOCAL SEARCH

RAOUF HAMZAOU<sup>\*</sup> AND DIETMAR SAUPE<sup>\*†</sup>

**Abstract.** Optimal fractal image compression is an NP-hard combinatorial optimization problem where the domain of feasible solutions is a large finite set  $\mathcal{T}$  of contractive affine mappings, and the cost function is  $\|f^* - f_T\|_2^2$ , where  $f^*$  is the original image, and  $f_T$  is the fixed point of  $T \in \mathcal{T}$ . In contrast, traditional fractal coders are based on a greedy algorithm known as collage coding, which minimizes  $\|f^* - T(f^*)\|_2^2$ . We describe a local search algorithm that rapidly improves the solution obtained by collage coding. In particular, we show how the successive computations of the cost function can be efficiently done by combining a Gauss-Seidel like iterative method and a graph algorithm.

**Key words.** Image compression, Combinatorial optimization, Local search, Graph algorithms, Iterative methods.

**AMS(MOS) subject classifications.** Primary 94A08, 90C27, 05C85, 65F10.

**1. Introduction.** Fractal image compression was introduced in the late eighties by Barnsley and Jacquin [2, 16]. The basic idea is to approximate a target digital image by the fixed point of a contractive affine transformation of a complete metric space of digital images. In spite of many improvements to the original approach [22], fractal coding has not reached the performance of the state-of-the-art in image compression. However, the potential of fractal image compression has not been fully exploited because current fractal schemes are based on a method called collage coding, which does not find the fixed point closest to the original image. Although the determination of one such point is an NP-hard problem [19, 13], better solutions than those obtained by collage coding were reported [4, 5, 3, 15, 7, 17, 21]. Apart from the method in [3, 17], all the others do not allow significant improvements over collage coding and are computationally expensive. In this paper, we present a local search algorithm [10–12] that provides a better solution than all previous techniques. Typical peak signal-to-noise ratio (PSNR) improvements over collage coding range from 0.2 dB to 0.8 dB for 8 bits per pixel (bpp)  $512 \times 512$  monochrome real-world images. We provide an efficient implementation of the algorithm and show in particular that the successive evaluations of the cost function, which requires computations of fixed points of high-dimensional affine transformations, can be simplified by combining a Gauss-Seidel like iterative method and a graph algorithm.

The paper is organized as follows. In Section 2, we introduce our terminology and explain the principles of fractal image compression. In

---

<sup>\*</sup>Institut für Informatik, Universität Leipzig, Augustusplatz 10–11, Leipzig, Germany (hamzaoui@informatik.uni-leipzig.de).

<sup>†</sup>saupe@informatik.uni-leipzig.de.

Section 3, we present previous work. In Section 4, we give our local search algorithm and provide an efficient implementation. Section 5 contains experimental results. In the last section, we discuss our results and suggest future work.

**2. Terminology.** We consider square monochrome digital images of size  $2^N \times 2^N$ ,  $N \geq 1$ , which will be seen as functions  $f : \mathcal{X} = \{0, \dots, 2^N - 1\} \times \{0, \dots, 2^N - 1\} \rightarrow \mathbb{R}$ . Thus, the intensity of pixel  $(i, j) \in \mathcal{X}$  is  $f(i, j)$ . Let  $\psi : \mathcal{X} \rightarrow \{1, \dots, 2^{2N}\}$  be a bijection called *pixel ordering*. Let  $B$  be a nonempty subset of  $\mathcal{X}$  containing  $n$  pixels. For a given  $\psi$ , we associate to  $f|_B$ , the restriction of  $f$  on  $B$ , an  $n$ -dimensional vector

$$\mathbf{x}_{\psi, f|_B} = (f\psi^{-1}(b_1), \dots, f\psi^{-1}(b_n))^T.$$

Here,  $T$  is the transpose symbol and  $\psi(B) = \{b_1, \dots, b_n\}$ , where  $b_1 < b_2 < \dots < b_n$ .

Let  $(\mathcal{F}, +, \cdot)$  be the vector space of digital images  $f : \mathcal{X} \rightarrow \mathbb{R}$ . Then  $\mathcal{F}$  is a Banach space for any norm. Given a target image  $f^*$ , a large finite set  $\mathcal{T}$  of contractive transformations  $T : \mathcal{F} \rightarrow \mathcal{F}$ , and a bit budget  $r$ , an optimal fractal encoding is a solution to the constrained minimization problem

$$\min_{T \in \mathcal{T}} \Delta(f^*, f_T) \text{ subject to } \text{len}(w(T)) \leq r.$$

Here  $f_T$  is the fixed point of  $T$ ,  $\Delta(f^*, f_T) \geq 0$  is the reconstruction error, typically  $\Delta(f^*, f_T) = \|f^* - f_T\|_2^2$ , and  $\text{len}(w(T))$  is the length of the codeword  $w(T)$  used to represent  $T$ . This codeword is sent to the decoder, which computes  $f_T$  as the limit point of the sequence of iterates  $\{f_k\}_{k \geq 0}$ , where  $f_{k+1} = T(f_k)$ , and  $f_0$  is an arbitrary starting image.

Let  $B = \{i, i+1, \dots, i+n-1\} \times \{j, j+1, \dots, j+n-1\} \subset \mathcal{X}$  be an  $n \times n$  square block. We denote by  $\mathbf{x}_{f|_B}$  the column vector formed by stacking the pixel intensities of  $B$  row by row, left to right, and top to bottom, that is,  $\mathbf{x}_{f|_B} = (f(i, j), f(i, j+1), \dots, f(i, j+n-1), f(i+1, j), f(i+1, j+1), \dots, f(i+n-1, j), f(i+n-1, j+1), \dots, f(i+n-1, j+n-1))^T$ . Let  $\mathcal{R} = \{R_1, \dots, R_{n_R}\}$  be a partition of  $\mathcal{X}$  into pairwise disjoint  $2^n \times 2^n$  square blocks called *range blocks*. Let  $\mathcal{D} = \{D_1, \dots, D_{n_D}\}$  be a set of  $2^{n+1} \times 2^{n+1}$  square blocks  $D_i \subset \mathcal{X}$  called *domain blocks*. Let  $\mathcal{S} = \{s_1, \dots, s_{n_s}\} \subset [-s_{\max}, s_{\max}]$ ,  $s_{\max} < 1$ , be a set of real numbers called *scaling factors*. Let  $\mathcal{O} = \{o_1, \dots, o_{n_o}\}$  be a set of real numbers called *offsets*. Let  $\mathcal{P} = \{\mathbf{P}_1, \dots, \mathbf{P}_{n_P}\}$  be a set of permutation matrices of order  $2^{2n}$ . We define a set  $\mathcal{T}$  of *fractal transforms*  $T : \mathcal{F} \rightarrow \mathcal{F}$ , where each  $T$  is given by an  $n_R$ -tuple of *fractal parameters*

$$(D_T(1), s_T(1), o_T(1), \mathbf{P}_T(1)), \dots, (D_T(n_R), s_T(n_R), o_T(n_R), \mathbf{P}_T(n_R))),$$

$(D_T(i), s_T(i), o_T(i), \mathbf{P}_T(i)) \in \Pi = \mathcal{D} \times \mathcal{S} \times \mathcal{O} \times \mathcal{P}$ , as follows. For each  $i \in \{1, \dots, n_R\}$  we have

$$\mathbf{x}_{T(f)|_{R_i}} = s_T(i) \mathbf{P}_T(i) \hat{\mathbf{D}} \mathbf{x}_{f|_{D_T(i)}} + o_T(i) \mathbf{1}.$$

Here  $\mathbf{1} = (1, \dots, 1)^T \in \mathbb{R}^{2^{2^n}}$ , and  $\hat{\mathbf{D}}$  is the  $2^{2^n} \times 2^{2^{(n+1)}}$  *downsampling matrix*

$$\hat{\mathbf{D}} = \frac{1}{4} \begin{pmatrix} \mathbf{Q} & \mathbf{Q} & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & \mathbf{Q} & \mathbf{Q} & 0 & \dots & 0 & 0 \\ \vdots & & & & & & & \vdots \\ 0 & \dots & 0 & \dots & 0 & \mathbf{Q} & \mathbf{Q} & 0 \end{pmatrix},$$

where  $\mathbf{Q}$  is the  $2^n \times 2^{n+1}$  submatrix

$$\mathbf{Q} = \begin{pmatrix} 1 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & \dots & 0 & 0 & 0 \\ \vdots & & & & & & & & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & 1 & 1 \end{pmatrix}.$$

The matrix  $\hat{\mathbf{D}}$  reduces the size of vector  $\mathbf{x}_{f|D_T(i)}$  to  $2^{2^n}$  by averaging the intensities of pairwise disjoint groups of four neighboring pixels of  $D_T(i)$ . The codeword of  $T$  consists of a sequence of bits that specify the fractal parameters of all range blocks. Because  $|s_T(i)| < 1$  for all  $i = 1, \dots, n_R$ ,  $T$  is a contraction for the  $l_\infty$  norm. Indeed, let  $f$  and  $g$  be two images. Then

$$\begin{aligned} & \|T(f) - T(g)\|_\infty \\ &= \max_{1 \leq i \leq n_R} \|\mathbf{x}_{T(f)|R_i} - \mathbf{x}_{T(g)|R_i}\|_\infty \\ &= \max_{1 \leq i \leq n_R} \|s_T(i) \mathbf{P}_T(i) \hat{\mathbf{D}}(\mathbf{x}_{f|D_T(i)} - \mathbf{x}_{g|D_T(i)})\|_\infty \\ &\leq \left( \max_{1 \leq i \leq n_R} |s_T(i)| \right) \max_{1 \leq i \leq n_R} \|\hat{\mathbf{D}}\|_\infty \|\mathbf{P}_T(i)\|_\infty \|\mathbf{x}_{f|D_T(i)} - \mathbf{x}_{g|D_T(i)}\|_\infty \\ &= \left( \max_{1 \leq i \leq n_R} |s_T(i)| \right) \max_{1 \leq i \leq n_R} \|\mathbf{x}_{f|D_T(i)} - \mathbf{x}_{g|D_T(i)}\|_\infty \\ &\leq \left( \max_{1 \leq i \leq n_R} |s_T(i)| \right) \|f - g\|_\infty. \end{aligned}$$

Thus, the decoding

$$(2.1) \quad f_0 \rightarrow T(f_0) \rightarrow T(T(f_0)) \rightarrow \dots$$

is convergent to the fixed point  $f_T$  independently of the initial image  $f_0$  (we say then convergent). In [9], it is shown that given a pixel ordering  $\psi$ , there exists a sparse matrix  $\mathbf{A}$  and a vector  $\mathbf{b}$  such that if  $\mathbf{x}_k = \mathbf{x}_{\psi, f_k|X}$ , then the decoding corresponds to the iterative method

$$(2.2) \quad \mathbf{x}_{k+1} = \mathbf{A} \mathbf{x}_k + \mathbf{b}.$$

A faster decoding

$$(2.3) \quad f_0 \rightarrow T_{GS}(f_0) \rightarrow T_{GS}(T_{GS}(f_0)) \rightarrow \dots$$

can be obtained from (2.1) by using the new pixel intensities as soon as they become available [9]. If the pixel intensities are computed in the order  $\psi^{-1}(1), \dots, \psi^{-1}(2^{2N})$ , then (2.3) corresponds to

$$(2.4) \quad x_u^{(k+1)} = \sum_{v \leq u-1} a_{u,v} x_v^{(k+1)} + \sum_{v \geq u} a_{u,v} x_v^{(k)} + b_u$$

for  $u = 1, \dots, 2^{2N}$ . Hamzaoui [9] shows that the Gauss-Seidel type iterative method (2.3) converges to  $f_T$ , and that if all scaling factors have the same sign, then the asymptotic rate of convergence of method (2.4) is equal to or larger than that of the conventional method (2.2).

**3. Previous work.** Let us assume that the codewords of the transformations  $T$  have the same length. Then an optimal transformation  $T_{\text{opt}}$  is one that minimizes the reconstruction error

$$E(T) = \Delta(f^*, f_T) = \|f^* - f_T\|_2^2$$

over all feasible solutions given by

$$((D_T(1), s_T(1), o_T(1), \mathbf{P}_T(1)), \dots, (D_T(n_R), s_T(n_R), o_T(n_R), \mathbf{P}_T(n_R))).$$

There are  $(n_D n_s n_o n_P)^{n_R}$  such solutions. Thus, finding  $T_{\text{opt}}$  by enumeration is impractical for large  $n_R$ . Usually, a suboptimal solution is found by a greedy algorithm known as collage coding [16]. The idea consists of minimizing the *collage error*  $\Delta(f^*, T(f^*))$  instead of the reconstruction error  $\Delta(f^*, f_T)$ . The motivation for collage coding is the inequality

$$\|f^* - f_T\|_2 \leq \frac{1}{1 - c(T)} \|f^* - T(f^*)\|_2,$$

where it is assumed that  $T$  is a contraction for the Euclidean metric, and where  $c(T)$  is the contraction factor of  $T$ . Collage coding simplifies the optimization problem because

$$\begin{aligned} \Delta(f^*, T(f^*)) &= \sum_{i=1}^{n_R} \|\mathbf{x}_{f^*|R_i} - \mathbf{x}_{T(f^*)|R_i}\|_2^2 \\ &= \sum_{i=1}^{n_R} \|\mathbf{x}_{f^*|R_i} - (s_T(i) \mathbf{P}_T(i) \hat{\mathbf{D}} \mathbf{x}_{f^*|D_T(i)} + o_T(i) \mathbf{1})\|_2^2. \end{aligned}$$

Hence optimal fractal parameters in collage coding are solutions of the  $n_R$  independent minimization problems

$$\min_{(D, s, o, \mathbf{P}) \in \Pi} \|\mathbf{x}_{f^*|R_i} - (s \mathbf{P} \hat{\mathbf{D}} \mathbf{x}_{f^*|D} + o \mathbf{1})\|_2^2, \quad i = 1, \dots, n_R.$$

Each of these minimization problems is solved as follows. For a given  $(D, \mathbf{P}) \in \mathcal{D} \times \mathcal{P}$ , let  $s$  and  $o$  denote the solutions of the least squares problem

$$\min_{s, o \in \mathbb{R}} \|\mathbf{x}_{f^*|R_i} - (s \mathbf{P} \hat{\mathbf{D}} \mathbf{x}_{f^*|D} + o \mathbf{1})\|_2^2.$$

If we denote the vector  $\mathbf{x}_{f^*|R_i}$  by  $\mathbf{r}$  and the vector  $\mathbf{P}\hat{\mathbf{D}}\mathbf{x}_{f^*|D}$  by  $\mathbf{c}$ , then the least squares solution is given by

$$(3.1) \quad s = \frac{2^{2n}(\mathbf{c}^T \mathbf{r}) - \mathbf{c}^T \mathbf{1} \mathbf{r}^T \mathbf{1}}{2^{2n} \mathbf{c}^T \mathbf{c} - (\mathbf{c}^T \mathbf{1})^2}$$

and

$$(3.2) \quad o = \frac{1}{2^{2n}}(\mathbf{r}^T \mathbf{1} - s \mathbf{c}^T \mathbf{1})$$

if  $\mathbf{x}_{f^*|D}$  is not in the linear span of  $\mathbf{1}$  and by  $s = 0$  and  $o = \frac{1}{2^{2n}} \mathbf{r}^T \mathbf{1}$ , otherwise. Next, we take the nearest neighbors of  $s$  and  $o$  in  $\mathcal{S}$  and  $\mathcal{O}$  respectively, yielding a scaling factor  $s^*$  and an offset  $o^*$ . Finally, we select a pair  $(D, \mathbf{P})$  in  $\mathcal{D} \times \mathcal{P}$  that minimizes the error

$$\|\mathbf{x}_{f^*|R_i} - (s^* \mathbf{P}\hat{\mathbf{D}}\mathbf{x}_{f^*|D} + o^* \mathbf{1})\|_2^2,$$

which is equal to

$$(3.3) \quad \mathbf{c}^T \mathbf{c} (s^*)^2 + 2\mathbf{c}^T \mathbf{1} s^* o^* + 2^{2n} (o^*)^2 - 2\mathbf{r}^T \mathbf{c} s^* - 2\mathbf{r}^T \mathbf{1} + \mathbf{r}^T \mathbf{r}.$$

The minimum such error is called collage error for range block  $R_i$ .

Several researchers recognized the suboptimality of collage coding and proposed better, though also suboptimal, solutions. Most of these works [4, 15, 7, 21] start from the solution found by collage coding, fix the domain blocks and the permutations, and optimize the scaling factors and the offsets (considered as continuous variables) by, for example, gradient descent methods, which yield local minima of the reconstruction error. However, after quantization, the PSNR improvement over collage coding is negligible for practical encodings [21]. Moreover, the time complexity of the optimization is too high. In contrast, Barthel and Voyé [3] and Lu [17] suggested to update all fractal parameters by an iterative procedure, which starts from an original solution  $T_0$  found by collage coding, then replaces at step  $n \geq 1$  the fractal parameters of all range blocks  $R_i$ ,  $i = 1, 2, \dots, n_R$ , by the solutions of the minimization problem

$$(3.4) \quad \min_{(D, s, o, \mathbf{P}) \in \Pi} \|\mathbf{x}_{f^*|R_i} - (s \mathbf{P}\hat{\mathbf{D}}\mathbf{x}_{f_{T_{n-1}}|D} + o \mathbf{1})\|_2^2.$$

In other words, at step  $n$ , one does collage coding based on the domain blocks with image intensities from the fixed point of step  $n-1$ . This method allows substantial PSNR improvements over collage coding. However, it has two drawbacks. First, there is no guarantee that the reconstruction error decreases after each step. Second, the procedure is time expensive because every step corresponds to a new encoding of the test image. To accelerate the procedure, Lu proposed to consider at each step only a portion of the range blocks, namely the range blocks  $R_i$  for which the ratio between the collage error and the reconstruction error  $\|\mathbf{x}_{f^*|R_i} - \mathbf{x}_{f_{T_{n-1}}|R_i}\|_2^2$  is largest.

**4. Local search.** Local search [18] was successfully used in many hard combinatorial optimization problems [1]. Given a set of feasible solutions  $\mathcal{T}$ , a neighborhood is defined as a function  $\mathcal{N} : \mathcal{T} \rightarrow 2^{\mathcal{T}}$ . The algorithm starts at some initial solution  $T \in \mathcal{T}$  and searches for a better solution in its neighborhood. As long as a better solution is found, it is adopted, and the search is repeated. The algorithm stops at  $T^*$ , a locally optimal solution with respect to the neighborhood. That is, if  $E$  is the cost function, then  $E(T^*) \leq E(T)$  for all  $T \in \mathcal{N}(T^*)$ . In our application, the neighborhood of a fractal transform  $T \in \mathcal{T}$  given by its fractal parameters  $(D_T(i), s_T(i), o_T(i), \mathbf{P}_T(i)) \in \Pi$ ,  $i = 1, \dots, n_R$ , will be the set of  $n_R$  transforms obtained from  $T$  by modifying the fractal parameters of only one range block  $R_r$ ,  $r \in \{1, \dots, n_R\}$ . This is done by setting the fractal parameters of  $R_r$  to the solution of the minimization problem

$$(4.1) \quad \min_{(D, s, o, \mathbf{P}) \in \Pi} \|\mathbf{x}_{f^*|R_r} - (s\mathbf{P}\hat{\mathbf{D}}\mathbf{x}_{f_T|D} + o\mathbf{1})\|_2^2.$$

The complete local search algorithm is as follows.

1. *Initialization.* Let  $M$  be a maximum number of trials. Set  $n := 0$ ,  $i := 0$ , and  $j := 0$ . Find an initial feasible solution  $T_0$  by collage coding. Let  $n_R$  be the number of range blocks in the partition.
2. Let  $r := 1 + (i \bmod n_R)$  and let  $T_c$  be the solution in the neighborhood of  $T_n$  obtained by modifying the fractal parameters of  $R_r$  according to (4.1). Set  $i := i + 1$ .
3. Compute the fixed point of  $T_c$ .
4. If  $E(T_c) < E(T_n)$ , set  $T_{n+1} := T_c$ ,  $n := n + 1$ ,  $j := 0$ . Otherwise set  $j := j + 1$ .
5. If  $(i \leq M$  and  $j < n_R)$  go to Step 2. Otherwise stop.

We now present several techniques for accelerating the above algorithm.

**4.1. No computation of fixed point.** In Step 2, if the new fractal parameters of range block  $R_r$  are equal to the current ones, then  $T_c = T_n$ . Thus, we just set  $j := j + 1$  and go to Step 5.

**4.2. Choice of starting point.** The most time-consuming part of the local search algorithm is the computation in Step 3 of the fixed point  $f_{T_c} = \lim_{k \rightarrow \infty} f_k$ , where  $\{f_k\}_{k \geq 0}$  is given by  $f_{k+1} = T_c(f_k)$ , and  $f_0$  is an arbitrary initial image. By taking  $f_0 = f_{T_n}$ , we have fast convergence because  $f_{T_n}$  is close to  $f_{T_c}$ .

**4.3. Ordering of the range blocks.** The local search algorithm is dependent on the ordering  $R_1, \dots, R_{n_R}$ . We found out that a good choice consists of indexing the range blocks according to decreasing error  $\|\mathbf{x}_{f^*|R_r} - \mathbf{x}_{f_{T_0}|R_r}\|_2^2$ .

**4.4. Decoding with dependence graph.** The main idea for accelerating the local search algorithm exploits the fact that  $T_c$  and  $T_n$  differ

only in the fractal parameters of range block  $R_r$ . Thus, in Step 3, if we start from the current fixed point  $f_{T_n}$  and apply the operator  $T_c$  once, then only the pixel intensities in  $R_r$  have to be updated, which avoids many unnecessary computations. If we now apply  $T_c$  to the first iterate  $T_c(f_{T_n})$ , then only the range blocks whose domain blocks overlap  $R_r$  have to be updated. Note that these range blocks may include  $R_r$ . This procedure is repeated until convergence to  $f_{T_c}$ . The decoding relation between the range blocks was studied by Domaszewicz and Vaishampayan [6] who introduced the notion of the *dependence graph* of a fractal transform.

DEFINITION 4.1. *We say that  $D \subset \mathcal{X}$  overlaps  $R \subset \mathcal{X}$  if  $D \cap R \neq \emptyset$ . Let  $T \in \mathcal{T}$  be a fractal transform. A range block  $R_j$  is called a child of a range block  $R_i$  if  $D_T(j) \cap R_i \neq \emptyset$ , that is if the domain block that encodes  $R_j$  overlaps  $R_i$ . The range block  $R_i$  is then called a parent of  $R_j$ . A dependence graph of  $T$  is a directed graph  $\mathcal{G}(T) = (V, E)$  where  $V$  is the image partition  $\mathcal{R}$ , and an ordered pair of vertices  $(R_i, R_j) \in E$  if  $R_j$  is a child of  $R_i$ .*

Note that each range block has at least one parent, but not all range blocks have children.

EXAMPLE 1. *Consider the image partition of Figure 1. Suppose that the set of domains  $\mathcal{D}$  consists of the following four blocks:  $D_1 = R_1 \cup R_2 \cup R_3 \cup R_4$ ,  $D_2 = R_5 \cup R_6 \cup R_7 \cup R_8$ ,  $D_3 = R_9 \cup R_{10} \cup R_{11} \cup R_{12}$ ,  $D_4 = R_{13} \cup R_{14} \cup R_{15} \cup R_{16}$ . Suppose that the domain blocks encode the range blocks as in Table 1. Then the dependence graph of the transform is given by Figure 2 (the figure was drawn with daVinci V2.1 [20]). Note that  $R_{13}, R_{14}, R_{15}$ , and  $R_{16}$  have no children because  $D_4$  is not used in the encoding.*

TABLE 1  
Range-Domain assignment.

$R_1$	$R_2$	$R_3$	$R_4$	$R_5$	$R_6$	$R_7$	$R_8$
$D_1$	$D_2$	$D_1$	$D_3$	$D_2$	$D_2$	$D_3$	$D_2$
$R_9$	$R_{10}$	$R_{11}$	$R_{12}$	$R_{13}$	$R_{14}$	$R_{15}$	$R_{16}$
$D_3$	$D_1$	$D_3$	$D_1$	$D_1$	$D_2$	$D_2$	$D_1$

The iteration scheme  $f_{T_n} \rightarrow T_c(f_{T_n}) \rightarrow T_c(T_c(f_{T_n})) \rightarrow \dots \rightarrow f_{T_c}$  can be implemented as a breadth-first traversal of the dependence graph of  $T_c$ , starting from vertex  $R_r$ . The first iteration  $f_{T_n} \rightarrow T_c(f_{T_n})$  corresponds to visiting the root vertex  $R_r$ , and for  $k \geq 2$ , iteration  $k$  corresponds to visiting the children of all vertices visited at iteration  $k - 1$ . Formally, if we denote by  $\{L_k\}_{k \geq 1}$  the sequence of subsets of  $\{R_1, \dots, R_{n_R}\}$  given by  $L_1 = \{R_r\}$  and  $L_{k+1} = \bigcup_{R_i \in L_k} \{R_j \in V : (R_i, R_j) \in E\}$ , then at iteration  $k$ , we compute the pixel intensities of only the range blocks in  $L_k$ . Note that only vertices that are connected to  $R_r$  are visited. Note also that in contrast to typical breadth-first search a vertex may be visited many times. However, at a given iteration, a vertex needs to be visited only once. In

$R_1$	$R_2$	$R_5$	$R_6$
$R_3$	$R_4$	$R_7$	$R_8$
$R_9$	$R_{10}$	$R_{13}$	$R_{14}$
$R_{11}$	$R_{12}$	$R_{15}$	$R_{16}$

FIG. 1. Image partition into 16 square range blocks.

fact, when visiting a vertex, we use the newly computed pixel intensities as soon as they are available. The proposed decoding corresponds then to the iteration scheme

$$(4.2) \quad f_{k+1} = T_{c,i_k}(f_k), \quad f_0 = f_{T_n},$$

where  $T_{c,i_k}$  is a Gauss-Seidel like operator (2.3) such that the pixels intensities of the range blocks in  $L_k$  are computed last. This iteration scheme converges to  $f_{T_c}$  as a consequence of the following proposition.

**PROPOSITION 4.1.** *Let  $T_c \in \mathcal{T}$  be the candidate transform, and let  $\mathcal{T}_c = \{T_{c,1}, \dots, T_{c,M}\}$ ,  $M \leq 2^{2N!}$ , be the set of all possible Gauss-Seidel like transformations derived from  $T_c$  by a reordering of the pixels. For  $k \geq 0$ , let  $i_k \in \{1, \dots, M\}$ . Then the iteration scheme  $f_{k+1} = T_{c,i_k}(f_k)$  converges to  $f_{T_c}$ .*

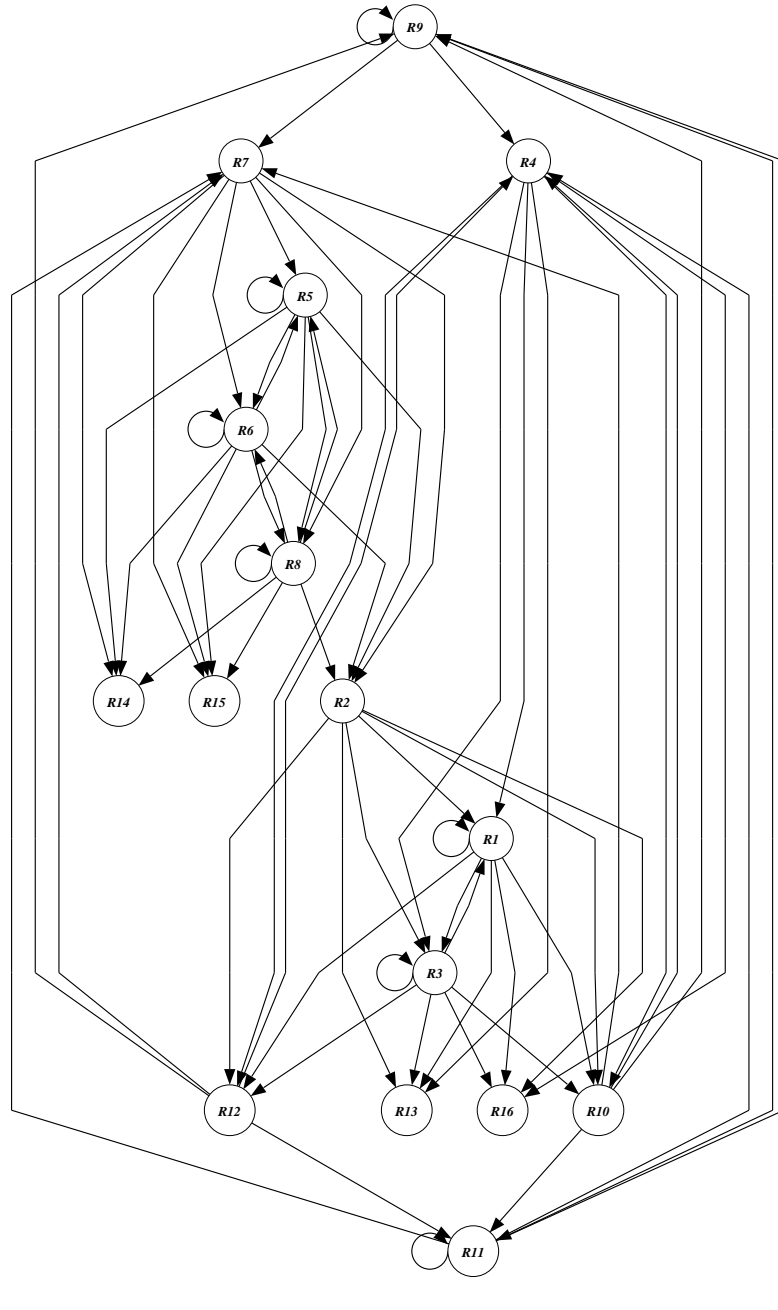
*Proof.* For all  $i = 1, \dots, M$ ,  $T_{c,i}$  is a contraction in the  $l_\infty$  norm with fixed point  $f_{T_c}$  [9]. Let  $t_i$  denote the contraction factor of  $T_{c,i}$  in  $l_\infty$ . Then

$$\begin{aligned} \|f_k - f_{T_c}\|_\infty &= \|T_{c,i_{k-1}}(f_{k-1}) - T_{c,i_{k-1}}(f_{T_c})\|_\infty \\ &\leq t_{i_{k-1}} \|f_{k-1} - f_{T_c}\|_\infty \\ &\leq t_{i_{k-1}} t_{i_{k-2}} \cdots t_{i_0} \|f_0 - f_{T_c}\|_\infty \\ &\leq \left( \max_{1 \leq i \leq M} t_i \right)^k \|f_0 - f_{T_c}\|_\infty. \end{aligned}$$

Thus, since  $(\max_{1 \leq i \leq M} t_i) < 1$ , we have  $\lim_{k \rightarrow \infty} f_k = f_{T_c}$ .  $\square$

The dependence graph is stored as an array of linked lists (adjacency lists), i.e., for each range block a linked list is kept, which holds all its children. Whenever the domain block that encodes range block  $R_r$  is changed, the dependence graph is updated as follows.





dar@inici.vz.1

FIG. 2. *Dependence graph.*

1. Determine the current parents of vertex  $R_r$ .
2. Go through the linked lists of these vertices and remove  $R_r$  from each list.
3. Determine the new parents of vertex  $R_r$ .
4. Insert  $R_r$  in the linked list of each new parent.

In example 1, suppose that  $R_r = R_1$ , and that  $R_r$  is now encoded by  $D_4$ . Then, the dependence graph is updated by removing  $R_1$  from the children lists of  $R_1, R_2, R_3$ , and  $R_4$  and inserting it in the lists of  $R_{13}, R_{14}, R_{15}$ , and  $R_{16}$ .

**4.5. Unchanged intensities in range and domain blocks.** The least squares approach used in Step 2 of the algorithm requires the computation of the sum of the pixel intensities and the sum of the squared pixel intensities of the original image in a range block (see (3.1), (3.2), and (3.3)). Because the same range block may be considered many times, one can save computing time by storing these sums during the initialization (Step 1). Similarly, the same sums have to be computed for  $f_{T_n}$  in all downsampled domain blocks. But according to Subsection 4.4, only the pixels of a few range blocks are updated (Table 2). Thus, for  $f_{T_{n+1}}$ , we recompute these sums only for the domain blocks that overlap these range blocks.

TABLE 2  
*Computation savings.*

Number $i$ of ranges	Number of ranges with unchanged parameters (4.1)	Average number of visited ranges (4.4)
$n_R = 4096$	1012 (24.71 %)	6.44
$2n_R = 8192$	3281 (40.05 %)	9.74

Finally, in Step 4, the test  $E(T_c) < E(T_n)$  reduces to

$$(4.3) \quad \sum_{R_i \in \bigcup_{k=1}^m L_k} \|\mathbf{x}_{f^*|R_i} - \mathbf{x}_{f_m|R_i}\|_2^2 - \|\mathbf{x}_{f^*|R_i} - \mathbf{x}_{f_{T_n}|R_i}\|_2^2 < 0,$$

where  $m$  is the iteration step at which (4.2) converged.

#### 4.6. Fast local search.

1. *Initialization.* Let  $M$  be a maximum number of trials. Set  $n := 0$ ,  $i := 0$ , and  $j := 0$ .
  - (a) Find an initial feasible solution  $T_0$  by collage coding. Let  $n_R$  be the number of range blocks in the partition.
  - (b) Compute and store the sum of the pixel intensities and the sum of the squared pixel intensities in each range block of the original image.
  - (c) Determine  $\mathcal{G}(T_0)$ , the dependence graph of  $T_0$ . Set  $\mathcal{G} = \mathcal{G}(T_0)$ .
  - (d) Sort  $R_1, \dots, R_{n_R}$  according to decreasing  $\|\mathbf{x}_{f^*|R_r} - \mathbf{x}_{f_{T_0}|R_r}\|_2^2$ .



FIG. 3. The  $512 \times 512$  Goldhill image.

2. Let  $r := 1 + (i \bmod n_R)$  and let  $T_c$  be the solution in the neighborhood of  $T_n$  obtained by modifying the fractal parameters of  $R_r$  according to (4.1). Set  $i := i + 1$ .
3. If  $T_c = T_n$ , set  $j := j + 1$  and go to step 7.
4. If  $D_{T_c}(r) \neq D_{T_n}(r)$ , update  $\mathcal{G}$ , i.e., set  $\mathcal{G} = \mathcal{G}(T_c)$ .
5. Use  $\mathcal{G}$  to compute the fixed point of  $T_c$ .
6. If (4.3) is true, set  $T_{n+1} := T_c, n := n + 1, j := 0$ . Otherwise set  $j := j + 1$  and  $\mathcal{G} = \mathcal{G}(T_n)$ .
7. If  $(i \leq M \text{ and } j < n_R)$ , go to Step 2. Otherwise stop.

**5. Experimental results.** The test image was the 8 bpp  $512 \times 512$  Goldhill image (Figure 3). The image support was partitioned into  $n_R = 4096$  range blocks of size  $8 \times 8$ . The domain blocks consisted of the  $n_D = 15625$  square blocks of size  $16 \times 16$  whose upper-left pixels are situated on locations  $(i, j)$ , where  $i \equiv 0 \pmod{4}$  and  $j \equiv 0 \pmod{4}$ . The parameters  $n_s$  and  $n_o$  were equal to 32 and 128, respectively. The only permutation matrix allowed was the identity. If for  $i \in \{1, \dots, n_R\}$  we had  $s_{T_0}(i) = 0$ , then the bits for the domain position were not stored because they are redundant. Note that our local search algorithm may assign a nonzero scaling factor to a range block for which collage coding yielded a

zero scaling factor. To ensure that our algorithm does not increase the size of the code, we do not modify the fractal parameters of such range blocks.

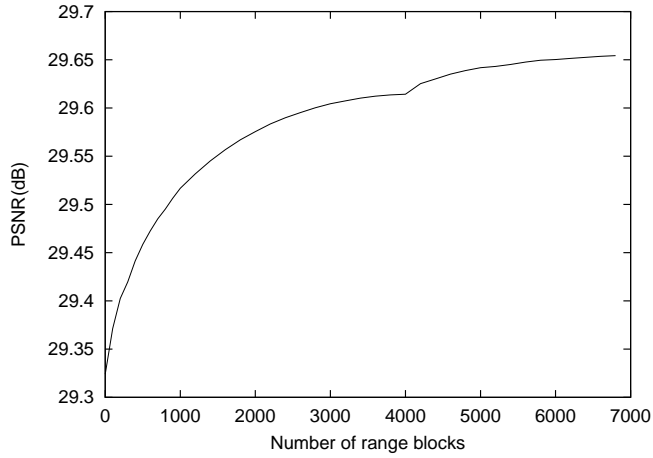
To compute the fixed point of  $T_c$ , we stopped the iteration (4.2) when the root of the mean-squared error between two consecutive image iterates was less than 0.1. Figure 4 shows the PSNR yielded by the solution  $T_n$  as a function (a) of the number of range blocks considered (counter  $i$  in the algorithm) and (b) CPU time. For an original 8 bpp image  $f^*$  and a decoded image  $f_T$  of size  $2^N \times 2^N$ , the PSNR in dB was computed as

$$\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\frac{1}{2^{2N}} \|f^* - f_T\|_2^2} \right).$$

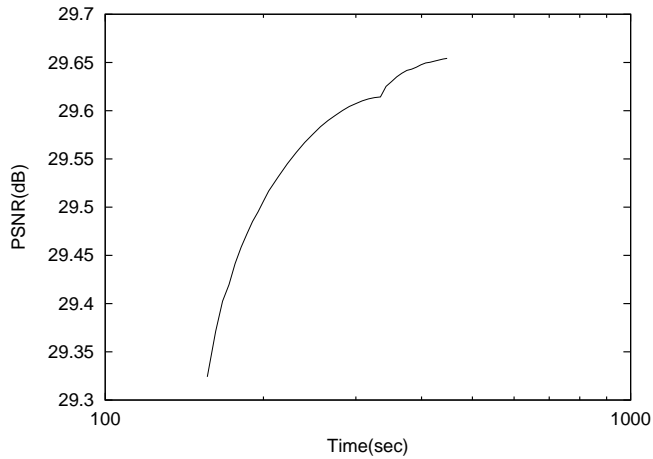
The CPU time was measured on an SGI Origin 200 with four 270 MHz MIPS R12000 processors and a main memory size of 1536 Megabytes. The initialization (collage coding) took 156 seconds and yielded a PSNR of 29.32 dB at a compression ratio of 19.68:1. The fast local search algorithm increased the PSNR by 0.3 dB in about 180 seconds. Table 2 illustrates the computation savings due to the techniques of 4.1 and 4.4. The first column gives the current number of range blocks considered (counter  $i$ ). The second column gives the number of blocks with unchanged fractal parameters. The last column gives the average number of range blocks that were visited in the breadth-first traversal of the dependence graph during the computation of  $f_T$ . For example, if after  $i$  steps,  $n_0$  vertices were visited, then the average number is  $\frac{n_0}{i-i_0}$ , where  $i_0$  is the number of steps in which the fractal parameters were unchanged. Finally, we point out that our algorithm used less than 0.6 Megabytes of extra space compared to collage coding. More details on memory requirements can be found in [12].

**6. Conclusion.** We showed how a local search algorithm can rapidly improve a solution found by collage coding. In particular, we solved the following problem. Given the fixed point of a fractal transform. How can one efficiently compute the fixed point of the fractal transform obtained by changing the parameters of only one range block? For simplicity, we described our algorithm for uniform image partitions. Results for more complex partitions [8, 14], which yield better rate-distortion performance can be found in [10–12].

Because we are not able to say how far our locally optimal solution is from the global one, the fundamental question of the potential of fractal image compression remains open. Topics for future research could be the determination of a better neighborhood function and the use of more sophisticated local search strategies that try to escape local minima by accepting neighbors that increase the cost function [1].



(a)



(b)

FIG. 4. (a) PSNR versus number of range blocks for the  $512 \times 512$  Goldhill image. (b) PSNR versus time (logarithmic scale) for the  $512 \times 512$  Goldhill image.

REFERENCES

[1] AARTS, E.H.L. AND LENSTRA, J.K. (eds.), *Local Search in Combinatorial Optimization*, Wiley, 1997.  
 [2] BARNSELY, M. AND HURD, L., *Fractal Image Compression*, AK Peters, Wellesley, 1993.  
 [3] BARTHEL, K. U. AND VOYÉ, T., *Adaptive fractal image coding in the frequency domain*, in: *Proc. Int. Workshop on Image Processing: Theory, Methodology, Systems and Applications*, Budapest, June 1994.

- [4] DOMASZEWICZ, J. AND VAISHAMPAYAN, V. A., *Structural limitations of self-affine and partially self-affine fractal compression*, in: *Proc. SPIE Visual Communications and Image Processing* (1993), **2094**: 1498–1504.
- [5] DOMASZEWICZ, J. AND VAISHAMPAYAN, V.A., *Iterative collage coding for fractal compression*, in: *Proc. ICIP-94 IEEE Int. Conf. on Image Processing*, Austin, Texas, Nov. 1994.
- [6] DOMASZEWICZ, J. AND VAISHAMPAYAN, V.A., *Graph-theoretical analysis of the fractal transform*, in: *Proc. ICASSP-1995 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Vol. 4, Detroit, 1995.
- [7] DUDBRIDGE, F. AND FISHER, Y., *Attractor optimization in fractal image encoding*, in: *Proc. of the Conference Fractals in Engineering*, Arcachon, June 1997.
- [8] FISHER, Y., *Fractal image compression with quadrees*, in: *Fractal Image Compression — Theory and Application*, Y. Fisher (ed.), Springer-Verlag, New York, 1994.
- [9] HAMZAOUI, R., *Fast iterative methods for fractal image compression*, *Journal of Mathematical Imaging and Vision* (1999), **11**(2): 147–159.
- [10] HAMZAOUI, R., HARTENSTEIN, H., AND SAUPE, D., *Local iterative improvement of fractal image codes*, *Image and Vision Computing* (2000), **18**: 565–568.
- [11] HAMZAOUI, R., SAUPE, D., AND HILLER, M., *Fast code enhancement with local search for fractal image compression*, in: *Proc. ICIP-2000 IEEE International Conference on Image Processing*, Vancouver, Sept. 2000.
- [12] HAMZAOUI, R., SAUPE, D., AND HILLER, M., *Distortion minimization with fast local search for fractal image compression*, *Journal of Visual Communication and Image Representation* (2001), **12**: 450–468.
- [13] HARTENSTEIN, H., *Topics in Fractal Image Compression and Near-Lossless Image Coding*, Doctoral Dissertation, University of Freiburg, 1998.
- [14] HARTENSTEIN, H., RUHL, M., AND SAUPE, D., *Region-based fractal image compression*, *IEEE Transactions on Image Processing* (2000), **9**(7): 1171–1184.
- [15] HÜRTGEN, B., *Performance bounds for fractal coding*, in: *Proc. ICASSP-1995 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Vol. 4, Detroit, 1995.
- [16] JACQUIN, A. E., *Image coding based on a fractal theory of iterated contractive image transformations*, *IEEE Trans. Image Processing* (1992), **1**: 18–30.
- [17] LU, N., *Fractal Imaging*, Academic Press, 1997.
- [18] PAPANIMITRIOU, C. H. AND STEIGLITZ, K., *Combinatorial Optimization: Algorithms and Complexity*, Dover, 1998.
- [19] RUHL, M. AND HARTENSTEIN, H., *Optimal fractal coding is NP-hard*, in: *Proc. DCC'97 Data Compression Conference*, J.A. Storer and M. Cohn (eds.), IEEE Comp. Soc. Press, March 1997.
- [20] daVinci V2.1, Computer Science Department, University of Bremen, Germany.
- [21] VRSCAY, E.R. AND SAUPE, D., *Can one break the “collage barrier” in fractal image coding*, in: *Fractals: Theory and Applications in Engineering*, M. Dekking, J.L. Veהל, E. Lutton and C. Tricot (eds.), pp. 307–323, Springer-Verlag, London, 1999.
- [22] WOHLBERG, B.E. AND DE JAGER G., *A review of the fractal image coding literature*, *IEEE Trans. Image Processing* (1999), **8**(12): 1716–1729.