# Local iterative improvement of fractal image codes

Raouf Hamzaoui [a] Hannes Hartenstein [b] Dietmar Saupe [a]

[a] *Institut für Informatik, Universität Leipzig, Augustusplatz 10–11, 04109 Leipzig, Germany*

[b] *Institut für Informatik, Universität Freiburg, Am Flughafen 17, 79110 Freiburg, Germany*

**Abstract**

In fractal image compression, the code is given by a contractive affine mapping whose fixed point is an approximation to the original image. Usually, the mapping is found by the collage coding method. We propose an algorithm that starts from an initial mapping obtained by collage coding and iteratively provides a sequence of contractive mappings whose fixed points are monotonically approaching the original image. Experimental results show that the rate-distortion improvement over collage coding is significant.

*Key words:* Fractal image compression; combinatorial optimization; local search

## 1 Introduction and previous work

In fractal image compression (6; 3), the code for an image $\mathbf{x}$ is an efficient binary representation of a contractive affine mapping $T$ whose unique fixed point (or *attractor*) $\mathbf{x}_T$ is a good approximation to $\mathbf{x}$. The decoding is based on the contraction mapping principle, which gives $\mathbf{x}_T$ as the limit point of the sequence of iterates $\{\mathbf{x}_n\}_{n\geq 0}$, where $\mathbf{x}_{n+1} = T(\mathbf{x}_n)$ and $\mathbf{x}_0$ is an arbitrary initial image.

In practice, the image is partitioned into nonoverlapping blocks (*range blocks*) $R_1, \ldots, R_{n_R}$. For each range block $R_i$, a transformation $W_i$ is found, which maps a larger block (*domain block*) $D_{d(i)}$, $d(i) \in \{1, \ldots, n_{D_i}\}$ of the same image to the range block. Here the blocks $R_i$ and $D_{d(i)}$ are considered as vectors of pixel intensities. The action of $W_i$ can be described as follows:

$$W_i(R_i) = s_{s(i)}\tau_{\tau(i)}AD_{d(i)} + o_{o(i)}\mathbf{1},$$

where $s_{s(i)} \in \mathbb{R}$, $s(i) \in \{1, \ldots, n_s\}$, is a scaling factor, $o_{o(i)} \in \mathbb{R}$, $o(i) \in \{1, \ldots, n_o\}$, is an offset, $A$ is the downsampling operator which shrinks the domain block via pixel averaging to match the range block size, $\tau_{\tau(i)}$, $\tau(i) \in \{1, \ldots, n_\tau\}$ is a permutation that shuffles the pixel intensities in the downsampled block, and $\mathbf{1}$ is the block with intensity 1 at every pixel. To ensure convergence of the decoding, the scaling factor is restricted to the interval $[-s_{\max}, s_{\max}]$, where $s_{\max} < 1$ (see (3) p. 51). The mapping $T$ is specified by the image partition and for each range block in the partition, the indexes of the scaling factor, the offset, the permutation, and the domain block. In the following, we call these indexes *fractal parameters.* Also we sometimes call the mapping $T$ a *fractal code.*

Let us suppose that the image partition is fixed. Then, an optimal fractal code $T_{\text{opt}}$ is one that minimizes the reconstruction error

$$E(T) = \|\mathbf{x} - \mathbf{x}_T\|_2^2$$

over all feasible configurations

$$((s(1), o(1), \tau(1), d(1)), \ldots, (s(n_R), o(n_R), \tau(n_R), d(n_R)))$$

of the fractal parameters. Since there is only a finite number $N$ of such configurations, $T_{\text{opt}}$ could in principle be determined by exhaustive enumeration. However, this approach is impractical because $N$ is exponential in $n_R$. In a particular case where strong restrictions are imposed on the coding settings (7), it is possible to find an optimal fractal code. But in this case, the rate-distortion performance of fractal image compression is unsatisfactory. In a general setting, it is unlikely that an algorithm that computes an optimal fractal code in reasonable time could ever be found (9). For this reason, almost all researchers use a technique known as *collage coding,* which yields only a suboptimal fractal code. In collage coding, the fractal parameters of a given range block $R_i$ are determined as follows (3) . For each domain block $D_m$ and each permutation $\tau_p$, the least squares problem

$$(s, o) = \arg\min_{s,o \in \mathbb{R}} \|R_i - (s\tau_p A D_m + o\mathbf{1})\|_2^2$$

is solved. The coefficient $s$ is clamped to $[-s_{\max}, s_{\max}]$ and then both $s$ and $o$ are uniformly quantized yielding $s_{m,p}$ and $o_{m,p}$. This gives a collage error

$$C_i(m, p) = \|R_i - (s_{m,p}\tau_p A D_m + o_{m,p}\mathbf{1})\|_2^2.$$

Finally, the domain block and permutation index pair minimizing the collage error is selected together with the corresponding scaling factor and offset indexes.

Barthel *et al.* (1) and Lu (8) proposed to improve the fractal code obtained by collage coding with the following algorithm.

**Algorithm BL**:

(1) **Initialization**: Let $m$ be a maximum number of iterations. Set $n := 0$. Find an initial fractal code $T_0$ by collage coding.
(2) Compute $\mathbf{x}_{T_n}$ and the reconstruction error $E(T_n)$.
(3) For each range block $R_i,\ i = 1, \ldots, n_R$ of the original image, determine a new set of fractal parameters by minimizing the collage error $C_i(m, p)$ now using the image blocks of *the attractor* $\mathbf{x}_{T_n}$ as feasible domain blocks. This gives a new fractal code $T_{n+1}$. Set $n := n+1$. If $n < m$, go to Step 2. Otherwise, stop.

The retained fractal code is the one (among $m$ produced) yielding the smallest reconstruction error. Note that the same image partition is used throughout the algorithm. Thus, the initial compression ratio is not modified.

Although successful, this method has some drawbacks. First, there is no guarantee that the reconstruction error is reduced at each iteration or at least that convergence occurs. Second, experimental results (see Section 3) show that the gain over collage coding can be minor. Finally, the algorithm is computationally expensive. To accelerate it, Barthel suggests in Step 3 to keep for a given range block the same domain block address, and to adjust only the other fractal parameters (1). For the same complexity reason, Lu (see (8) p. 83) proposes to update only a small portion of the range blocks, namely the 10 % that have the highest ratio between their reconstruction error and the corresponding collage error.

## 2 Iterative improvement

In Algorithm BL, at each iteration the fractal parameters of all range blocks are updated. Instead, we propose to modify at an iteration step the fractal parameters of only one range block. Moreover, we update the current fractal code only if the new fractal code decreases the reconstruction error $E(T)$. This gives the following algorithm.

**Proposed Algorithm**:

(1) **Initialization**: Let $M$ be a maximum number of trials. Set $n := 0,\ i := 0$ and $k := 0$. Find an initial fractal code $T_0$ by collage coding. Compute the attractor of the fractal code $T_0$. Let $n_R$ be the number of range blocks in the partition.
(2) Let $r := 1 + (i \bmod n_R)$. Take range block $R_r$ from the original image. Determine for this range block new fractal parameters by minimizing the collage error $C_r(m, p)$ for the domain blocks of the attractor $\mathbf{x}_{T_n}$. Set $i := i + 1$.
(3) Construct a candidate fractal code $T_c$ by changing only the fractal parameters of range block $R_r$ according to the result of Step 2. Compute the attractor of the candidate fractal code and the reconstruction error $E(T_c)$.
(4) If $E(T_c) < E(T_n)$, set $T_{n+1} := T_c, n := n + 1, k := 0$. Otherwise set $k := k + 1$.
(5) If $(i \leq M$ and $k < n_R)$ go to Step 2. Otherwise stop.

The algorithm is successful in practice because in the first iterations the probability that a candidate fractal code decreases the current reconstruction error is high. Moreover, the rate of decrease of the reconstruction error is largest at the beginning. Thus, a satisfactory improvement is already obtained after a few updates.

## 3 Experimental results

Tables 1 and 2 compare the performance of collage coding, Algorithm BL and our proposed algorithm for several gray scale (eight bits per pixel) images. The reconstruction error was measured with the peak signal-to-noise ratio (PSNR) defined for $n \times n$ images by

$$\mathrm{PSNR} = 10 \log_{10}\left(\frac{255^2}{\frac{1}{n^2}\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2}\right)$$

where $\mathbf{x}$ and $\hat{\mathbf{x}}$ are the original image and the reconstructed image, respectively. Note, however, that the PSNR is not a perfect measure of image quality. Thus an increase of the PSNR may not necessarily be accompanied by an improvement in perceived image quality.

The encoding was based on Fisher's quadtree coder (4). At each quadtree level, domain blocks had twice the linear size of the range blocks, and their upper-left pixels were situated on a lattice with a spacing of $d = 8$ pixels. Full search was used, that is, all domain blocks on the lattice were inspected. The parameters $n_s$, $n_o$, and $n_T$ were equal to 32, 128, and 8, respectively. Table 1 shows results for uniform partitions into 8 × 8 range blocks. Table 2 shows results for a four-level quadtree partition, where the largest range block size was $32 \times 32$ and the smallest range block size was $4 \times 4$. The root-mean-square (rms) threshold was set to 18. In Algorithm BL, $m$ was set to 10. Since for this algorithm the reconstruction error is not a decreasing function, it is not clear if more iterations would provide better results. In our algorithm, $M$ was set to $2n_R$ for the 256 × 256 images, to 2000 for the 512 × 512 image with the four-level quadtree and to $n_R$ for the 512 × 512 image with the uniform partition.

In Step 3, the attractor of the candidate fractal code $T_c$ was efficiently constructed by starting the iterations from the attractor of the current fractal code $T_n$ and using a Gauss-Seidel like technique (5). In this way, a satisfactory approximation of the attractor $\mathbf{x}_{T_c}$ was obtained after only 2 iterations.

| Image | $n_R$ | Compression ratio | Collage coding PSNR (dB) | Algorithm BL PSNR (dB) | Proposed PSNR (dB) |
|---|---|---|---|---|---|
| 256 × 256 Lenna | 1024 | 20.45:1 | 26.51 | 26.63 | 26.77 |
| 256 × 256 San Francisco | 1024 | 20.45:1 | 24.54 | 24.65 | 24.92 |
| 512 × 512 Boat | 4096 | 18.96:1 | 29.74 | 29.87 | 30.01 |

Table 1
Reconstruction error in PSNR for uniform 8 × 8 partitions.

| Image | $n_R$ | Compression ratio | Collage coding PSNR (dB) | Algorithm BL PSNR (dB) | Proposed PSNR (dB) |
|---|---|---|---|---|---|
| 256 × 256 Lenna | 817 | 24.94:1 | 25.91 | 26.33 | 26.58 |
| 256 × 256 San Francisco | 1387 | 14.80:1 | 26.11 | 26.85 | 26.98 |
| 512 × 512 Boat | 1603 | 46.88:1 | 26.87 | 27.42 | 27.55 |

Table 2
Reconstruction error in PSNR for four-level quadtree partitions.

In many cases, the attractor generated by the proposed algorithm had a clearly better perceptual quality than the one obtained from collage coding. This is illustrated in Figures 1 and 2. Here the 256 × 256 Lenna image was encoded as in Table 2 with the exception of the rms threshold, which was set to 14 and the lattice spacing $d$, which was equal to 4 pixels.

(a)

(b) (c)

Fig. 1. Comparison of decoded images at a compression ratio of 17.62:1. (a) original 256 × 256 Lenna image (b) collage coding: PSNR = 27.57 dB (c) proposed algorithm: PSNR = 28.27 dB.

## 4 Discussion and future work

The proposed algorithm enhanced collage coding. The gain in image fidelity was up to 0.8 dB for the same compression ratio. With respect to Algorithm BL (1; 8), the improvement in PSNR was in some cases substantial, e.g., 0.27 dB for the San Francisco image.

(a)

(b)

Fig. 2. Zoomed images. (a) collage coding (b) proposed algorithm.

We conclude with some topics for further research. Though important, the gain in rate-distortion performance over collage coding was somehow penalized by an increase in encoding time. Including fast searching techniques (10) in Step 2 of the algorithm can be useful. Additional speed up is expected in Step 3 by exploiting the *dependence graph* of the fractal code (2). For example, if the range block $R_r$ is not overlapped by a domain block used in the fractal code $T_n$, then the attractor of $T_c$ can be computed from the current attractor $\mathbf{x}_{T_n}$ in a straightforward manner by updating the pixel intensities of only $R_r$.

Finally, it would be interesting to see if similar gains are obtained when our algorithm is used with more adaptive image partitions (e.g., rectangular), which enable a better rate-distortion performance than quadtrees.


## Acknowledgements

We thank Michael Hiller for implementing the algorithms.


## References


[1] K. U. Barthel and T. Voyé, Adaptive fractal image coding in the frequency domain, in: *Proceedings International Workshop on Image*

*Processing: Theory, Methodology, Systems and Applications* (Budapest, June 1994) 33–38.

[2] J. Domaszewicz and V. A. Vaishampayan, Graph-theoretical analysis of the fractal transform, in: *Proceedings ICASSP'1995 IEEE International Conference on Acoustics, Speech and Signal Processing* (Detroit, 1995) Vol. 4, 2559–2562.

[3] Y. Fisher, *Fractal Image Compression — Theory and Application* (Springer-Verlag, New York, 1994).

[4] Y. Fisher, Fractal image compression with quadtrees, in: Y. Fisher (ed.), *Fractal Image Compression — Theory and Application* (Springer-Verlag, New York, 1994).

[5] R. Hamzaoui, Fast decoding algorithms for fractal image compression, Technical Report 86, Institut für Informatik, Universität Freiburg, March 1997.

[6] A. E. Jacquin, Image coding based on a fractal theory of iterated contractive image transformations, *IEEE Trans. Image Processing* 1 (1992) 18–30.

[7] S. Lepsøy, G. E. Øien and T. Ramstad, Attractor image compression with a fast non-iterative decoding algorithm, in: *Proceedings ICASSP'1993 IEEE International Conference on Acoustics, Speech and Signal Processing* (1993) Vol. 5, 337–340.

[8] N. Lu, *Fractal Imaging* (Academic Press, 1997).

[9] M. Ruhl and H. Hartenstein, Optimal fractal coding is NP-hard, in: J. A. Storer and M. Cohn (eds.), *Proceedings DCC'97 Data Compression Conference* (IEEE Comp. Soc. Press, 1997) 261–270.

[10] D. Saupe, Fractal image compression via nearest neighbor search, in: Y. Fisher (ed.), *Proceedings NATO ASI Fractal Image Encoding and Analysis Trondheim, July 1995* (Springer-Verlag, Berlin Heidelberg, 1998) 95–115.