COST-BASED REGION GROWING FOR FRACTAL IMAGE COMPRESSION

Hannes Hartenstein, Dietmar Saupe
Institut für Informatik, Universtität Freiburg,
Am Flughafen 17, 79110 Freiburg, Germany
Tel: +49 761 203 8185; fax: +49 761 203 8182
e-mail: hartenst,saupe@informatik.uni-freiburg.de

ABSTRACT

For application in fractal coding we investigate image partitionings that are derived by a merge process starting with a uniform partition. At each merging step one would like to opt for the rate-distortion optimal choice. Unfortunately, this is computationally infeasible when efficient coders for the partition information are employed. Therefore, one has to use a model for estimating the coding costs. We discuss merging criteria that depend on variance or collage error and on the Euclidean length of the partition boundaries. Preliminary tests indicate that improved coding costs estimators may be of crucial importance for the success of our approach.

1 INTRODUCTION

In fractal image compression the image to be coded is partitioned into image blocks called *ranges*. Each range is approximated by another image part, called *domain*, which is scaled geometrically and modified using an affine transformation for the grey values. The fractal code consists of the partitioning information and, for each range, the domain block indices and the transformation parameters. Under certain constraints this code represents a contractive transformation whose unique fixed point is an approximation to the original image.

One of the bit allocation problems of fractal coding is to decide how many bits should be spent for the partitioning information. The most widely used strategies to partition the image are hierarchical methods like quadtree or horizontal/vertical partitioning. Here the coding costs for the partition are relatively low. On the other hand, more complex partitionings could lead to lower approximation errors. A region-growing based approach was introduced to fractal compression in [1]. This approach was extended in [2, 3] leading to a fast fractal coder with coding results comparable to Fisher's HV-coder [4].

Image partitioning problems can be stated as optimization problems where a specific cost function has to be minimized [5]. In this paper we present a formulation of the partitioning problem in fractal coding along these lines and analyze two cost functions that include the segmentation coding costs. The study was motivated by the question whether adding a segmentation cost constraint to the cost function will improve coding results, and by the need to determine properties of *good* partitionings.

This paper is organized as follows: In Section 2 we give basic definitions and state the partitioning problem as an optimization problem. We also comment on the inherent complexity of this problem. In Section 3 we outline the partition coding scheme used in this study. In Sections 4 and 5 algorithms for region-based fractal coding are presented. We report on coding results in Section 6 and give examples of decoded images.

2 OPTIMIZATION PROBLEM

Let $I : S = \{0, ..., 2^n - 1\} \times \{0, ..., 2^n - 1\} \to \{0, ..., 255\}$ be an image, and let $\hat{I}$ be the image of size $2^{n-1} \times 2^{n-1}$ that is obtained from I via pixel averaging, i.e., $\hat{I}(i, j) = \sum_{0 \le k,m \le 1} I(2i + k, 2j + m)$.

A partition $\mathcal{P}$ for I is a set of subsets of S such that $\bigcup_{P \in \mathcal{P}} = S$, $P \cap Q = \emptyset$ for $P, Q \in \mathcal{P}, P \neq Q$, and each $P \in \mathcal{P}$ is connected in the 4-neighbor sense. I restricted to a region $P \in \mathcal{P}$ is called a range of I.

Given a partition $\mathcal{P}$, the fractal code for image I is determined as follows. For each range R of this partition a block D_R of $\hat{I}$ (or of $\hat{I}$ under a square isometry) and real parameters s_R, o_R are sought such that

$$\|R - s_R D_R - o_R \mathbf{1}\|^2 = \min_{D} \min_{|s|<1, o \in \mathbb{R}} \|R - s^q D - o^q \mathbf{1}\|^2.$$

Here, $\|\cdot\|$ stands for the Euclidean norm, $\mathbf{1}$ is a vector with a 1 in each component, and s^q, o^q stand for the quantized *luminance* parameters. If R corresponds to region P we set $E(P) := \|R - s_R D_R - o_R \mathbf{1}\|^2$, and the overall approximation error, called *collage error*, is defined by

$$E(\mathcal{P}) := \sum_{P \in \mathcal{P}} E(P).$$

The global optimization problem is to find in the space of all partitions the optimal one in the sense that the rate of the corresponding fractal code is lower than a given budget limitation while simultaneously the overall collage error is minimized. Thus, one is faced with the following optimization problem:

$$\min_{\mathcal{P}} E(\mathcal{P}) + \lambda[\text{costs}(\mathcal{P}) + \text{costs}(code)],$$

where the total rate is split into the costs for coding the partition and the costs for coding the luminance parameters and domain indices. λ is a scale parameter that is used to control the compression ratio. When the number of bits for s, o are fixed, costs($code$) only depends on the number of ranges. But even when the coding costs are neglected, the optimization is infeasible as can be seen by its relation to the problem of piecewise constant approximations. Let assume that all scaling factors have to be zero. Then each block is approximated by the mean value of that block. The problem of finding the partitioning with a certain number of blocks that give minimal distortion is NP-hard [7]. This indicates the inherent complexity of our optimization problem. Thus, there is a need for good heuristic methods.

3 REGION EDGE MAP ENCODING

For the encoding of the partition information we use the *region edge map encoding technique* of Tate [8]. Recently, this coding scheme was successfully applied in a lossless coder for palettized images [9]. Assume that the partition to be coded is derived by a merging process starting with a uniform partition that consists of atomic square blocks. The region edge map (REM) has 2 bits for each atomic block indicating whether there is an edge to the north, and edge to the west, edges to the north and to the west, or no edge at all. The REM is encoded using an context-based adaptive arithmetic coder. The context for pixel $x_{i,j}$ consists of the four pixels $x_{i,j-1}, x_{i-1,j-1}, x_{i-1,j}$, and $x_{i-1,j+1}$. Due to the small size of the underlying alphabet, no context quantization is necessary. REM encoding gives superior performance compared to chain coders. Unfortunately, it is generally hard to estimate the change in coding costs when two ranges are merged because of the adaptive probability modeling. We therefore use the Euclidean length of the partition boundaries as an estimate for the actual partition coding costs. While the Euclidean length does not give good estimates for partitions that are *close* to uniform partitions, it can be used as a guidance for partitions corresponding to high compression ratios (see Figure 1).

4 VARIANCE-BASED REGION GROWING

When one employs a region growing method, one has to decide on the merge criterion. Assume that one starts with a uniform partitioning of, e.g., 4×4 blocks. In the context of fractal image compression, a pair of neighboring ranges should be merged when for the resulting range a domain block can be found that gives only a small approximation error. However, it is computation-

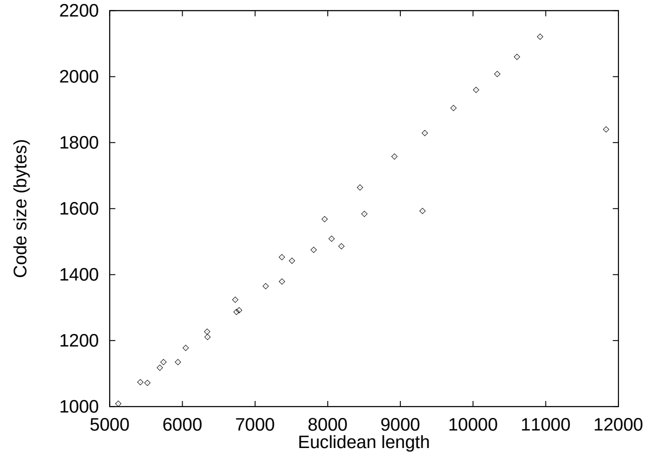


Figure 1: This figure shows the dependency between the Euclidean length of the partition boundaries and the actual coding costs using the REM scheme for various encodings.

ally infeasible to perform an optimal-domain-search for every pair of ranges that can be merged.

In this section we feature a region growing method where the merge criterion is based on the variance of the ranges [5, 6]. Therefore, we separate the image partitioning step from the actual fractal coding, i.e., the partition is derived as a preprocessing step and in a second step the optimal domain and luminance parameters are determined for each range.

The cost function to be minimized is

$$\sum_{P \in \mathcal{P}} [\text{size}(P) \cdot \text{Var}(P)] + \lambda \cdot \text{len}(\mathcal{P}),$$

where size(P) gives the number of pixels in this block, len($\mathcal{P}$) gives the Euclidean length of the partitioning boundaries, and Var(P) stands for the variance of the block P. Although the rate of a fractal code depends on the luminance parameters and the domain indices as well as on the partitioning coding costs, we can neglect the costs for those parameters in the cost function since we assume the same costs for each range. Therefore, any merger of two ranges will reduce the coding costs for the luminance parameters and the domain indices by the same amount. The **variance-based region growing** algorithm of [5, 6] proceeds as follows:

1. Start with a uniform partition of 4×4 blocks and $\lambda = 0$.

2. Merge all pairs of regions that decrease the cost function.

3. Increase the scale parameter λ and go to 2 or stop.

Of course, the outcome of this algorithm depends on the actual order in which ranges are merged in step 2. For the resulting partitioning we determine a fractal code by means of the FFT-based fractal encoding method [10, 11]. By incorporating the hardware-optimized Fast

Fourier Transform function of the Complib library [12] one obtains an additional speed-up factor of about 2.5 compared to the results of [10].

5 COLLAGE-BASED REGION GROWING

We now would like the merging criterion to be based on the collage-error. Thus, two questions arise that are intertwined with each other: how to maintain a fractal code during the merging process, and how to decide what neighboring range pair should be merged next? Since it is computational infeasible to perform after each merger a full search for all range pairs that include the newly merged range, we can only allow a restricted search. In order to search at promising positions, we maintain an *extended* fractal code, i.e., a fractal code that contains d domain addresses for each range. The **collage-based region growing I** [3] now works as follows:

1. Start with a uniform partitioning of, e.g., 4×4 blocks; for each region the best d domain blocks are determined.

2. Merge the pair that gives the least increase in collage error when approximated by a domain that is on one of the $2d$ positions that are optimal for the 2 ranges.

3. Stop for a given number of ranges.

This algorithm gives state-of-the-art coding results for fractal coders (with respect to rate-distortion performance *and* encoding times) when the initial block size is adjusted according to the desired compression ratio. However, the algorithm does not take into account the coding costs of the partitioning. Therefore, we have modified the algorithm by adopting ideas of the variance-based region growing algorithm of Section 4. Thus, in order to minimize the cost function

$$E(\mathcal{P}) + \lambda \cdot \text{len}(\mathcal{P})$$

steps 2 and 3 are changed. In the **collage-based region growing II** method, they are defined as follows:

2*. Merge all pairs that decrease the cost function when approximated by a domain that is on one of the $2d$ positions that are optimal for the 2 ranges.

3*. Increase the scale parameter λ and goto 2^* or stop.

The merging order is based on a greedy strategy: at each step the range pair is combined whose merger gives the largest decrease of the cost function.

6 EXPERIMENTS

A comparison of the compression ratio vs PSNR performance of the three region growing based fractal coding schemes are given in Figure 2. The variance-based algorithm is comparable to Fishers quadtree based coder

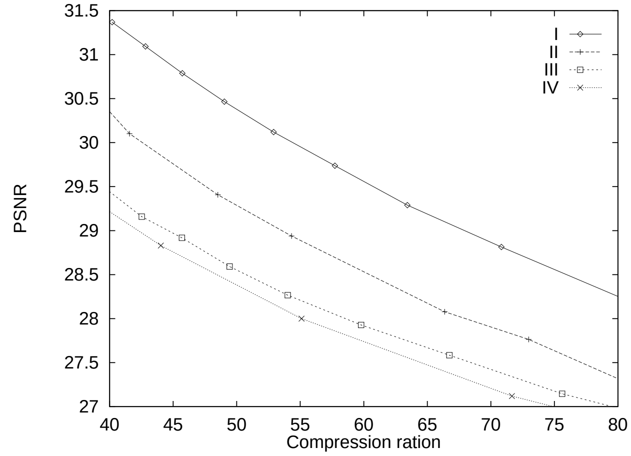
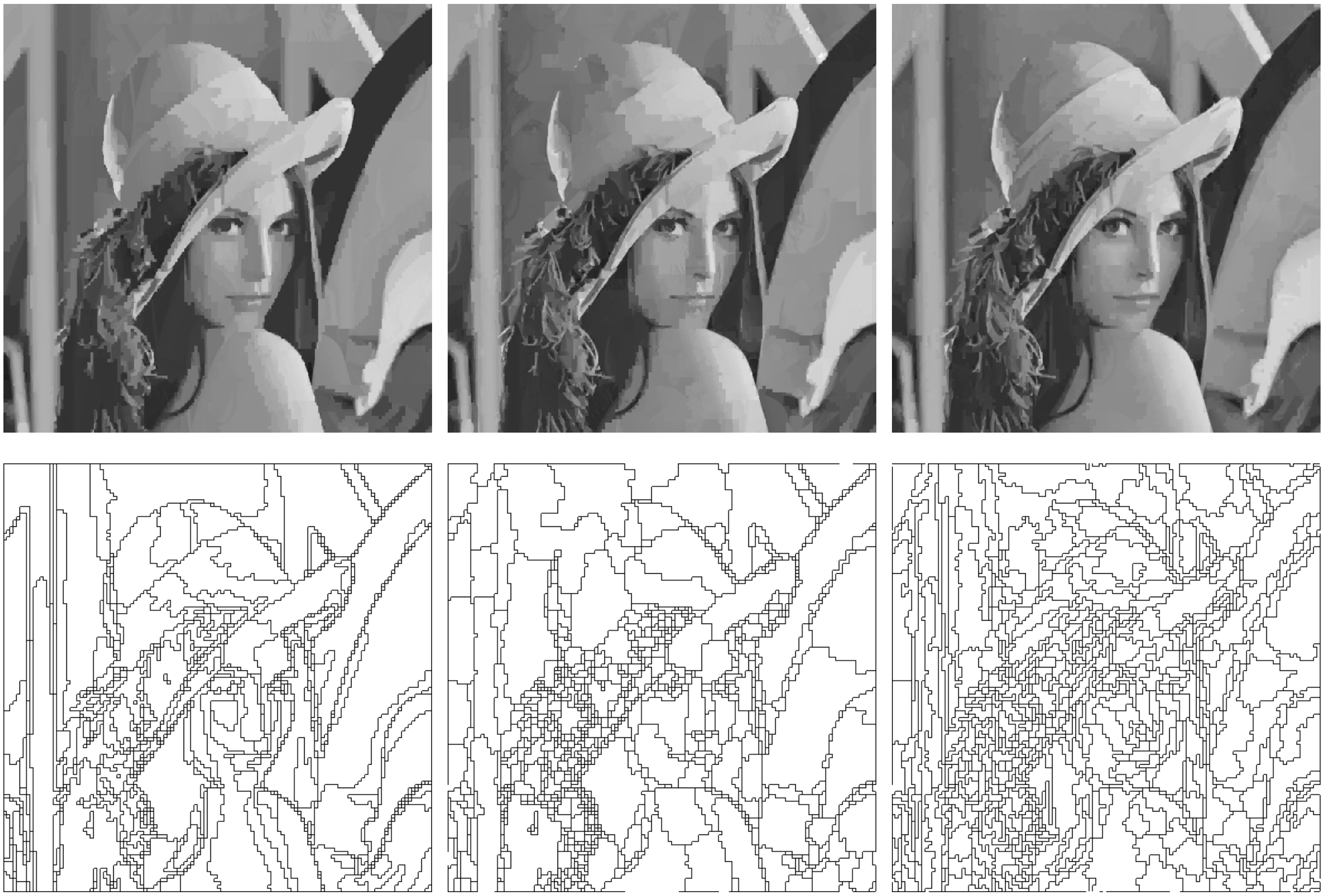


Figure 2: Compression ratio vs PSNR performance of the three region growing based methods and the quadtree-based method of [4] for the standard 512×512 Lenna image. I: collage-based method I, II: collage-based method II, III: variance-based method, IV: quadtree-based method.

with regard to PSNR performance. The collage based region growing II is better than the variance-based scheme. However, the incorporation of the segmentation length into the cost function does not lead to a better rate-distortion curve compared to the algorithm that is blind on the segmentation costs. With the variance-based algorithm one obtains many large ranges that degenerate to almost constant areas after decoding (see Figure 3(a)). On the other hand, there are a lot of small range blocks at edges although those edges could be approximated well by other image parts. This shows that the variance is an inadequate feature to derive partitionings by region growing for fractal coding. This is in contrast to quadtree-based fractal coding where it has been shown that the variance-based quadtree partition gives the same rate-distortion performance than the quadtree partition that is based on the actual collage error [13]. Comparing the images given in Figure 3 one notices that range blocks should not grow too large since otherwise the decoded image will show a lot of blocking artifacts. The partition of Fig. 3 (a) has Euclidean length 6046 and the partition coding costs represent 26.5% of the total rate, whereas the partition of Fig. 3 (c) has length 8725, representing 44% of the total rate.

7 CONCLUSION & FUTURE WORK

Since finding the optimal partitioning for a given bit rate is an intractable problem one has to be satisfied with heuristic methods. We have reported on methods for region growing that incorporate the length of the partition boundaries as an estimate for the actual segmentation coding costs. However, we could not improve the current state-of-the-art region based fractal coder. Two reasons are responsible for this: firstly, one obtains many large range blocks that lead to blocking artifacts



a) variance-based b) collage-based II c) collage-based I

Figure 3: The 512 × 512 image Lenna compressed at a ratio of about 66:1 and the corresponding partitions.

in the decoded images, and secondly, the segmentation length is an inadequate measure for the real segmentation coding cost. Work in the future should address the problem of how to get more accurate estimates of the actual partition coding costs.

References

[1] Thomas, L., Deravi, F., *Region-based fractal image compression using heuristic search,* IEEE Trans. Image Proc. 4,6 (1995) 832–838.

[2] Saupe, D., Ruhl, M., *Evolutionary fractal image compression,* Proc. ICIP '96, Lausanne, Sept. 1996.

[3] Ruhl, M., Hartenstein, H., Saupe, D., *Adaptive partitionings for fractal image compression*, Proc. ICIP'97, Santa Barbara, Oct. 1997.

[4] Fisher, Y., *Fractal Image Compression — Theory and Applications*, Springer-Verlag, New York, 1994.

[5] Morel, J., Solimini, S., *Variational Methods in Image Segmentation*, Birkhäuser, Boston, 1995.

[6] MegaWave2 Software, CEREMADE, Université de Paris IX-Dauphine, France

[7] Asano, T., Chen, D. Z., Katoh, N., Tokuyama, T., *Polynomial-time solutions to image segmentation*, Proc. 7th ACM-SIAM Symposium on Discrete Algorithms, Jan. 1996.

[8] Tate, S. R., *Lossless compression of region edge maps*, Technical Report CS-1992-9, Dept. of Computer Science, Duke University, 1992

[9] Ausbeck, P., *Context models for palette images*, Proc. IEEE Data Compression Conference, Snowbird, Utah, 1998

[10] Saupe, D. Hartenstein, H., *Lossless acceleration of fractal image compression by fast convolution*, Proc. ICIP'96, Lausanne, Sept. 1996.

[11] Hartenstein, H., Saupe, D., *Lossless acceleration of fractal image encoding via the Fast Fourier Transform*, submitted

[12] Complib, Silicon Graphics Scientific Mathematical Library, Silicon Graphics Inc.

[13] Saupe, D., Jacob, S., *Variance-based quadtrees in fractal image compression*, Electronic Letters 33, 1 (1997), 46–48.