ADAPTIVE POST-PROCESSING FOR FRACTAL IMAGE COMPRESSION

Nguyen Ky Giang, Dietmar Saupe

Universität Leipzig, Institut für Informatik, Augustusplatz 10–11, 04109 Leipzig, Germany

ABSTRACT

Fractal image compression is a block based technique. The blocks of the underlying image partition are called range blocks. Like all block based techniques, fractal coding suffers from blocking artifacts, in particular at high compression ratios. Moreover, due to the recursive nature of the fractal decoder blocking artifacts also show up *in the interior* of the range blocks. This paper proposes an effective post-processing to reduce both kinds of these artifacts. We apply a smoothing filter adapted to the partition before each iteration of the decoding process which reduces the blocking artifacts in the interior of the range blocks. After decoding the reconstructed image is enhanced by applying another adaptive filter along the range block boundaries. This filter is designed to reduce blocking artifacts while maintaining edges and texture of the original image. The experimental results show that the artifacts are reduced and significant improvements can be achieved. Our method compares favourably with previously published post-processing techniques for fractal image compression.

1. INTRODUCTION

The literature in the field of post-processing of compressed images is large. In particular, coding artifacts due to block based DCT as in JPEG have been studied extensively. Generally, post-processing can be seen either as image enhancement or as image restoration [9]. In the case of image restoration, the post-processing is considered as a reconstruction problem. Some classical methods, i.e., projection onto convex sets (POCS) [10], constrained least squares (CLS) [11], require a prior distortion model and thus cannot be straightforwardly adapted to fractal compression. Image enhancement methods are performed based on the local contents of the decoded image and are heuristic in the sense that no objective criterion is optimized. A straightforward approach is to apply lowpass filtering to the region, where artifacts occur [1, 6, 8]. Naturally, the filter must be adapted to the block boundaries, i.e., the image partition. If in addition high frequency details such as edges and texture are to be preserved, then lowpass filter coefficients should be selected appropriately in regions where such details occur. Therefore, some adaptive spatial filtering methods [2, 7] have been proposed to deal with this problem.

In fractal image compression [1] a partition of the image into so-called range blocks is required. Examples are the *quadtree partition* [1] and *the split-and-merge partition* [4, 5]. We assume that the reader is familiar with the concept of fractal image coding. Generally, in block based coding schemes, blocks are coded independently, and, thus, there are ”discontinuities” along adjacent blocks causing annoying blocking artifacts. These artifacts become worse as the compression ratio increases.

Fractal image compression is one of many block-based techniques. However, its code is self-referential which distinguishes fractal coding from all other approaches. This self-referential character has consequences for the blocking artifacts and poses special requirements for post-processing techniques that have been overlooked in the previous literature. In fractal image compression the blocking artifacts appear not only at the range boundaries, but also inside of the ranges because of the iterative decoding process, although at a less severe degree. This is due to the self-referential fractal code which requires the decoder to be iterative. Figure 1 illustrates, how artificial details inside the ranges come into existence. The range R_1 is best approximated by the domain D_1. The domain D_1 is covered by four range blocks. Assume there are discontinuities due to blocking artifacts at the boundaries of these four adjacent blocks. These artificial edges are copied by the iterative decoding process to the corresponding locations inside the range R_1.

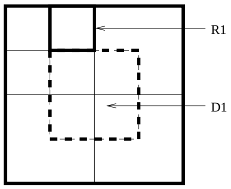

Figure 1: Range and corresponding domain.

In all proposed post-processing methods for fractal image compression, the problem of artificial details inside the ranges as explained above was not addressed. In this paper we introduce a post-processing technique which solves this problem by exploiting precisely that mechanism of fractal coding that also causes these artificial details. In a second step we develop a new signal adaptive post-processing filter that reduces the common blocking artifacts while maintaining natural edges and texture.

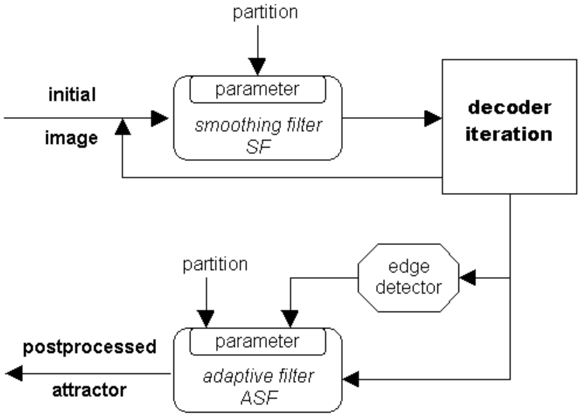

Figure 2: The post-processing scheme.

2. THE PROPOSED METHOD

There are two conflicting goals for the desired post-processing. The first is to eliminate the visible blocking artifacts along range block boundaries and also inside the range blocks. The second goal is to prevent deterioration of natural details such as edges and texture along range block boundaries. Figure 2 illustrates the pro-

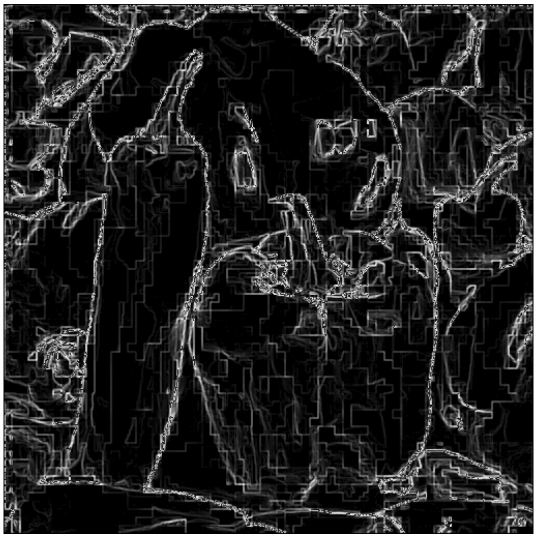

Figure 3: Gradient image of reconstructed Peppers 512x512, compression ratio 76:1, adaptive split-and-merge partition, block size 8, 800 ranges, threshold 10.

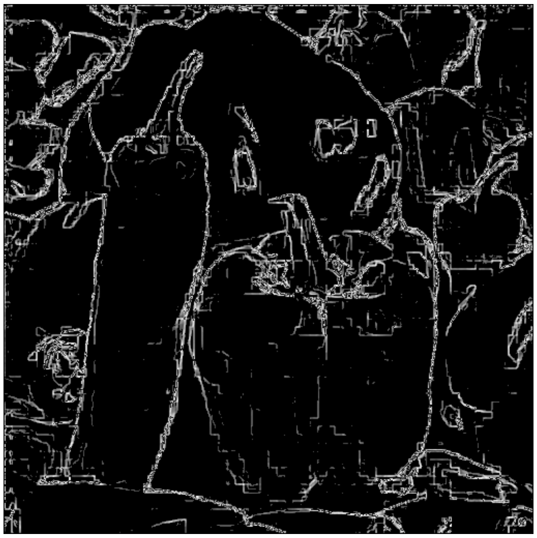

Figure 5: Gradient image of reconstructed Peppers 512x512, compression ratio 76:1, adaptive split-and-merge partition, block size 8, 800 ranges, threshold 85.

posed post-processing method. In standard fractal decoding one starts decoding with an arbitrary initial image (e.g., a blank image) and obtains the attractor after some iterations, say 6 to 10. Conventional post-processing would then commence after termination of the decoder iterations. In contract, we propose to apply a *smoothing filter SF* along range block boundaries *before* each iteration of the decoder. This implicitly changes the fractal image operator and its fixed point. The purpose is to reduce the artificial details inside range blocks as explained in Section 1. The integration of the post-processing into the decoder has the advantage that the degree of the smoothing inside the range blocks is automatically adjusted to the severeness of the targeted artifacts since both artifacts and smoothing effects are governed by the same scaling factors of the fractal code.

After the decoding phase with integrated smoothing severe blocking artifacts still remain at the range block boundaries. It would be possible to apply the filter SF one more time to reduce these artifacts. However, superior performance can be achieved by an *adaptive filter ASF*. We propose to adapt this filter to the local structure of the decoded image in order not to also smooth edges and texture from the original image.

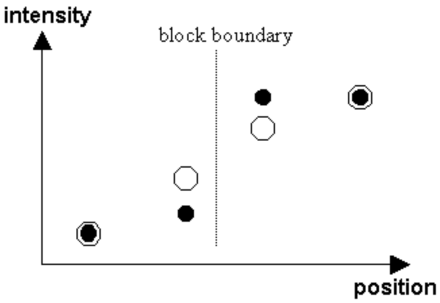

Figure 4: Four pixels from a scan line crossing a block boundary before (dots) and after filtering (circles) by SF.

We now present the details of the proposed smoothing filters. Previous methods in [1] and [2] modify the pixel intensities at range block boundaries using a weighted average of their values. We adapt the method in [1] for the smoothing filter SF. If the pixel values on either side of a boundary are a and b, then a is replaced by $w_1 \cdot a + w_2 \cdot b$, and b is replaced by $w_2 \cdot a + w_1 \cdot b$. The weighting factors w_1 and w_2 are non-negative and sum to 1, $w_1 + w_2 = 1$, see Figure 4. For the smoothing filter SF we use $w_1 = \frac{3}{4}$ and $w_2 = \frac{1}{4}$. These values are heuristic, derived from some probing experiments. In practice it suffices to apply SF only for the last 3 or 4 decoder iterations.

The adaptive filter after decoder iteration is based on the edge information of the attractor, estimated by the Sobel operators [3],

$$S_h = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad S_v = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}.$$

We illustrate the motivation for our approach outlined below by two experiments with results shown in Figures 3, 5, 6. First, we fractally encode a test image, Peppers, and generate the attractor in the standard way, without any post-processing. Then we evaluate the sum of the absolute values $|S_h(x)| + |S_v(x)|$ at all pixels x in the attractor image and show the results as two thresholded images using thresholds of 10 and 85 for the sum in Figures 3 and 5, respectively. Clearly, the blocking artifacts stand out clearly in Figure 3, but not so much in Figure 5. We conclude that large values of Sobel operators typically occur at natural edges of the image while edges due to blocking artifacts have smaller values. In a second experiment we concentrate on range block boundaries and evaluate an edge indicator $S(x)$ defined by $|S_h(x)|$ at pixels x along horizontal range boundaries, $|S_v(x)|$ at pixels x along vertical boundaries, and $\max(|S_h(x)|, |S_v(x)|)$ at pixels x that simultaneously lie at a horizontal and vertical boundary (corner pixels). Histograms for the edge indicator $S(x)$ are provided for the original test image Lena and for the attractor image (without post-processing, compression ratio 76:1) in Figure 6. We observe from the histograms that blocking artifacts cause artificial edges with values of $S(x)$ up to about 85. This fact leads to the approach to adapt the smoothing filter so that it applies stronger smoothing at pixels x with low edge indicator and less smoothing at pixels with higher values, where original edges are more likely. Thus, along horizontal range block boundaries, for the pixel above and below the boundary, we propose to apply an adaptive smoothing

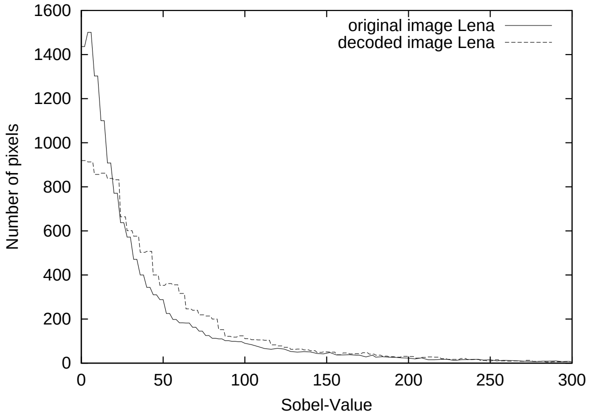

Figure 6: Histograms of $S(x)$ for original Lena test image and attractor image, compression ratio 76:1, no post-processing.

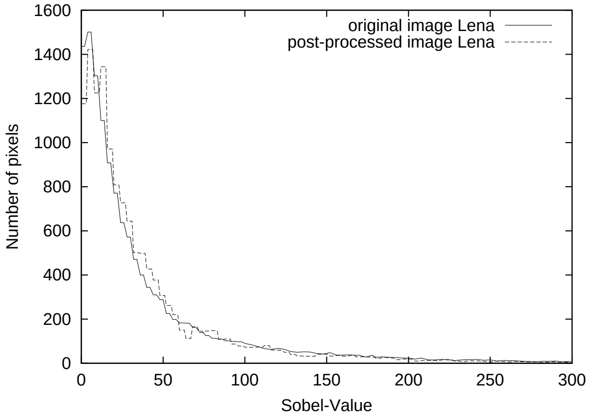

Figure 7: Histograms of $S(x)$ for original Lena test image and attractor image, compression ratio 76:1, with post-processing.

filter ASF similar to SF as before, but with weights

$$w_1 = \begin{cases} 2/3 & : \text{ if } |S_h(x)| < 85 \\ 3/4 & : \text{ if } 85 \leq |S_h(x)| < 160 \\ 5/6 & : \text{ if } 160 \leq |S_h(x)| < 350 \\ 1 & : \text{ if } 350 \leq |S_h(x)| \end{cases}$$

and $w_2 = 1 - w_1$. The process is likewise for vertical range boundaries. Special cases at the crossing are treated specially. There are three cases of crossing, see Figure 8. The pixels marked by circles are treated normal as explained above. For pixels shown as dots, we compute the absolute values $|S_h(x)| + |S_v(x)|$ and apply lowpass filtering of size 3x3, if the computed value falls below a threshold of $t = 350$.

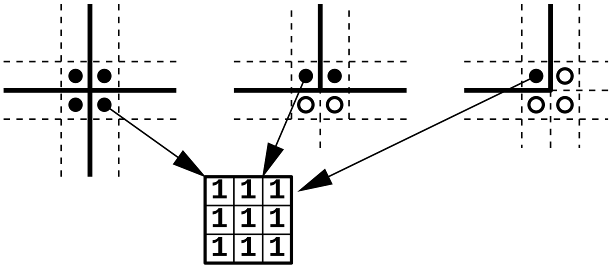

Figure 8: Cases for corner pixels.

rate	b	r	PSNR	FI	NL	SF	SF+ASF
0.084	8	600	27.92	0.32	0.00	0.10	0.41
0.105	8	800	28.65	0.33	-0.01	0.10	0.43
0.173	8	1500	29.98	0.34	-0.07	0.08	0.44
0.155	4	800	29.44	0.27	-0.06	0.12	0.44
0.205	4	1200	30.79	0.18	-0.16	0.11	0.43
0.302	4	2000	32.55	-0.12	-0.43	0.05	0.26

Table 1: Enhancement (dB) for image Lena 512x512 fractal coded at various bit rates (bpp) using adaptive split-and-merge scheme, (b unit block size, r number of ranges). See Figure 9 and 10 for images illustrating the improvements corresponding to the second row of the table.

rate	b	r	PSNR	FI	NL	SF	SF+ASF
0.084	8	600	27.36	0.44	0.04	0.14	0.50
0.105	8	800	28.33	0.45	0.04	0.13	0.53
0.173	8	1500	30.00	0.37	0.00	0.10	0.48
0.155	4	800	29.21	0.37	0.02	0.09	0.52
0.205	4	1200	30.67	0.25	-0.03	0.08	0.47
0.302	4	2000	32.35	-0.07	-0.12	0.02	0.27

Table 2: Enhancement (dB) for image Peppers 512x512 fractal coded at various bit rates (bpp) using adaptive split-and-merge scheme.

rate	PSNR	FI	SF	SF+ASF
0.23	28.24	0.41	0.11	0.56
0.18	27.62	0.43	0.13	0.58

Table 3: Enhancement (dB) for image Lena 512x512 fractal coded at various bit rates (bpp) using quadtree scheme.

rate	PSNR	FI	SF	SF+ASF
0.23	28.29	0.40	0.13	0.55
0.18	27.56	0.43	0.14	0.59

Table 4: Enhancement (dB) for image Peppers 512x512 fractal coded at various bit rates (bpp) using quadtree scheme.

In the sense of post-processing classification our adaptive filter is similar to other post-processing techniques [1, 2, 6, 7, 8], i.e., its strategy and functionality are based on the local content of the decoded image. But altogether, our adaptive filter exploits more local properties of the reconstructed image and therefore enhances it better in both reducing blocking artifacts and preserving high details.

3. EXPERIMENTAL RESULTS AND CONCLUSION

We have applied our filtering method to a number of different fractally coded images using the quadtree [1] and the split-and-merge [4, 5] partitions. In all cases, the blocking artifacts were significantly reduced and better results in terms of PSNR were achieved. The measured PSNR are given in Tables 1, 2, 3, 4 and the post-processing enhanced image of Lena is shown in Figure 10. FI is the method introduced in [1], NL the method in [2], computed using threshold 8.

After the smoothing filter SF was used during decoding, we

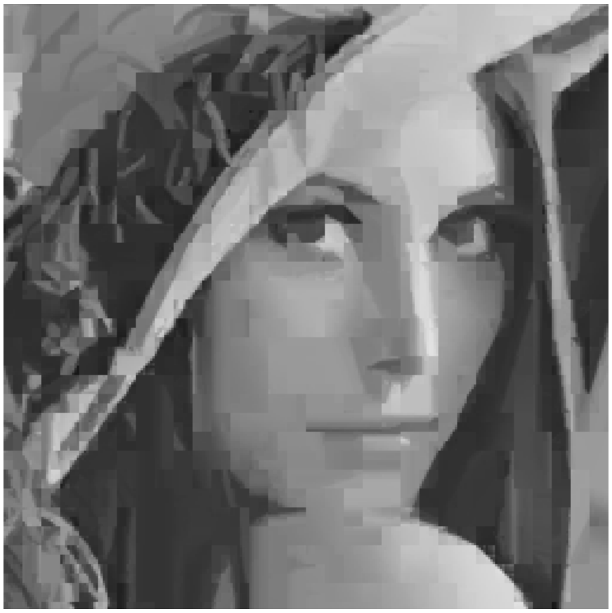

Figure 9: Zoomed decompressed image Lena, compression ratio 76:1, bit rate 0.105 bpp, PSNR 28.65 dB.

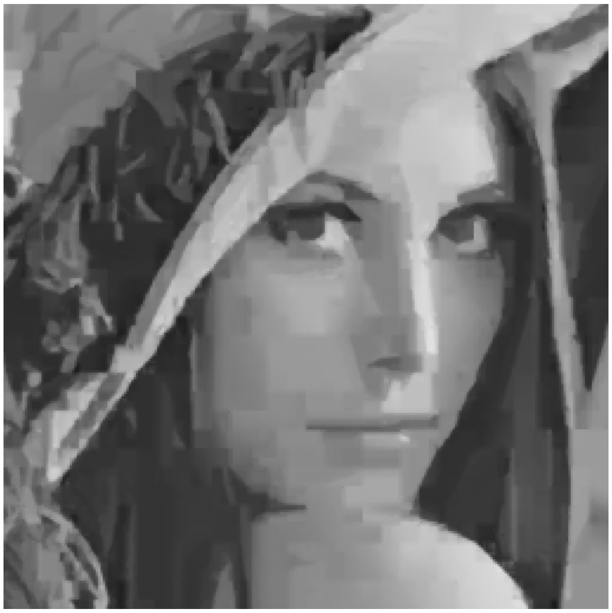

Figure 10: Zoomed decompressed image Lena with proposed post-processing, PSNR 29.08 dB, compare second row in Table 1.

obtained attractors with PSNR improvements of about 0.10 dB due to reduced artifacts in the interior of the ranges. All conventional post-processing techniques apply filtering only at range block boundaries and, thus, cannot improve the decoded image in the interior of range blocks as our method does. An extra improvement of 0.1 dB may seem small overall, but compared to 0.4 dB which is the maximum of post-processing benefits to be obtained from filtering at range block boundaries only once after decoder termination it is a significant, 25%, improvement.

Our second, adaptive filter smoothes the reconstructed image preserving important original edge details and does not blur the reconstructed image. In cases of low compression ratios, where the quality of the reconstructed image is high, the conventional methods FI and NL fail due to blurring while our method still achieves good performance. Note that the improvements in PSNR achieved by our proposed method were nearly constant independent of the compression ratio and chosen unit block sizes. Also visually, the post-processing significantly enhanced the attractor images. We show in Figure 7 that the post-processing re-adjusted the histogram of $S(x)$ as anticipated. In summary, we have proposed an efficient, yet simple post-processing scheme, which is adapted to the unique characteristics of fractal compression and significantly enhances the reconstructed image.

4. REFERENCES

[1] Y. Fisher, ”Fractal Image Compression Theory and Application”, Springer Verlag, 1994.

[2] N. Lu, ”Fractal Imaging”, Academic Press, 1997.

[3] R. C. Gonzalez, R. E. Woods, ”Digital Image Processing”, Addition-Wesley, 1992.

[4] M. Ruhl, H. Hartenstein, D. Saupe, ”Adaptive partitioning for fractal image compression”, Proc. IEEE ICIP-97, Santa Barbara, Oct. 1997.

[5] H. Hartenstein, M. Ruhl, D. Saupe, ”Region-based fractal image compression”, IEEE Transactions on Image Processing, in press.

[6] H. Reeve and J. Lim, ”Reduction of blocking effects in image coding”, Optical Engineering 23, pp. 34-37, Jan./Feb. 1984.

[7] B. Ramamurthi, A. Gersho, ”Nonlinear space-variant post processing of block coded image”, IEEE Transaction on Acoustic, Speech and Signal Processing, vol. ASSP-34, pp. 1258-1268, Oct. 1986.

[8] Y.-C. Chang, B.-K. Shyu, C.-Y. Cho, J.-S. Wang, ”Adaptive Post-Processing for Region-Based Fractal Image Compression”, Proc. DCC’2000, J. A. Storer and M. Cohn (eds.), IEEE Comp. Soc. Press, pp. 549, March 2000.

[9] M.-Y. Shen, C.-C. Kuo, ”Review of post-processing techniques for compression artifact removal”, Journal of Visual Communication and Image Representation, vol. 9, no.1, pp. 2-14, 1998.

[10] A. Zachor, ”Iterative procedures for reduction of blocking effects in transform image coding”, IEEE Trans. Circuits Syst. Video Technol. 2, pp. 91-95, Mar. 1992.

[11] Y. Yang, N. Galatsanos, A. Katsaggelos, ”Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images”, IEEE Trans. Circuits Syst. Video Technol. 3, pp. 421-432, Dec. 1993.