COMPARISON OF MEAN-SQUARE AND ABSOLUTE VALUE DISTORTION MEASURES IN FRACTAL CODING OF STILL IMAGES

F.C. Cesbron and F.J. Malassenet
Georgia Tech Lorraine
The European Platform of the Georgia Institute of Technology
2-3, rue Marconi
F–57070 Metz, France
Tel: (33) 87 20 39 39; fax: (33) 87 20 39 40
e-mail:fcesbron@georgiatech-metz.fr, fjm@georgiatech-metz.fr
(Project co-funded by Région de Lorraine, France)

ABSTRACT

Fractal coding often blurs or smoothes images. In particular, edges are poorly coded due to the choice of the distortion measure. Indeed, mean square error possesses no edge preserving property. The solution proposed in this article is to code using fractals and to introduce the L_1 norm or the absolute value distortion measure. The visual quality of the coded images with the L_1 norm is improved. However, the computational complexity of the coder is drastically increased. It may be kept low by preprocessing using an edge detection scheme to select the pertinent measure. The reconstruction algorithm remains the same.

1 INTRODUCTION

Automatic fractal coding algorithms on still images were first introduced by Barnsley and Jacquin [1] and improved by others, see [2] for example. The main idea is to model one block (range block) of an image with an affine transformation of another one (domain block) taken from the image itself. The matching domain block and transformation minimize the mean square error between the range block and the transformed domain block.

However, the coded image is often a blurred, due to the choice of the distortion measure. The mean square error possesses no edge preserving property and does not correspond to the human perception [3, 4]. One solution, proposed by Jacquin in [1] is block classification, with respect to geometric characteristics: shade, midrange, or edge blocks. Only the class which the range block belongs to is searched. Edges are better coded and the computation time is reduced. Fisher proposes another direction of research in [2]: detection and separate coding of chains of edges.

The solution presented here is to use the L_1 norm or absolute value distortion measure.

2 FRACTAL CODING

2.1 General Notations

The size of the original image is $M \times M$ pixels. The range block size is $N \times N$ pixels, denoted r_i, with $i = 0, 1, ..., N^2 - 1$. Domain blocks are blocks of original size $2N \times 2N$ pixels, decimated by a factor of two in both directions. Pixels in the decimated domain blocks are denoted d_i, with $i = 0, 1, ..., N^2 - 1$. The scaling factor, α, and the offset, t, define the affine transformation. The maximum allowed scaling factor is α_{max}. Convergence of the reconstruction is guaranteed for $\alpha_{max} < 1$. However, for improved results, $\alpha_{max} < 2$ is often used. For each range block, all the domain blocks are considered and the one with the smallest error is the matching one.

2.2 L_2 Norm

The error to minimize is:

$$e_2 = \min_{\alpha,t} \sum_{i=0}^{N^2-1} (\alpha d_i + t - r_i)^2$$

subject to

$$|\alpha| \leq \alpha_{max}.$$

Since e_2 is a differentiable function, this is a simple minimization problem, solved by setting the partial derivatives of e_2 with respect to α and t to zero [2].

2.3 L_1 Norm

The error to minimize is:

$$e_1 = \min_{\alpha,t} \sum_{i=0}^{N^2-1} |\alpha d_i + t - r_i|$$

subject to

$$|\alpha| \leq \alpha_{max}.$$

2.3.1 Linear Programming Problem

Since the function e_1 is not differentiable, the problem and its constraint are first linearized to obtain the equivalent problem [5]:

$$\min \sum_{i=0}^{N^2-1} (u_i + v_i) + M.(v_1'' + v_2'')$$

subject to

$$\begin{aligned} r_i = (a_1 - a_2).d_i + (b_1 - b_2) + u_i - v_i \\ a_1 - a_2 + u_1'' - v_1'' = \alpha_{max} \\ -a_1 + a_2 + u_2'' - v_2'' = \alpha_{max} \\ a_1, a_2, b_1, b_2, u_1'', u_2'', v_1'', v_2'', u_i, v_i \geq 0. \end{aligned}$$

where $\alpha = a_1 - a_2$, $t = b_1 - b_2$, M is any large number, and $i = 0, 1, ..., N^2 - 1$.

2.3.2 Modification of the Simplex Algorithm

The constraints define a convex set. The solution of the minimization problem is one of its vertices. The Simplex algorithm allows to go from one vertex to another, until the minimum is reached, as described for example in [6].

The modification of the Simplex algorithm, developed in [5, 7], passes through several vertices in a single iteration. The minimum is achieved using fewer iterations and less memory.

3 EDGE DETECTION

3.1 Complexity Comparison

Since every (range, domain) block pair can be considered independently of the others, the numbers of operations are evaluated for every such pair. Moreover, this algorithm may be implemented in parallel.

For every (range, domain) block pair, the L_2 algorithm, as implemented, requires approximately

N^2	adds or subtractions
N^2	multiplies or divisions

and the L_1 algorithm requires

$3 \cdot i \cdot N^2$	adds or subtractions
$(6 + s) \cdot i \cdot N^2$	multiplies or divisions

where i is the number of iterations and s the number of vertices passed through in one iteration. For a 256×256 image, with range block size 8×8, the average values are for $i = 3$ and $s = 9$. As a consequence, the computation time for the L_1 algorithm is 25 times longer than the L_2 algorithm.

3.2 Gradient Operators

To reduce the computational burden, edges are first detected in the original image, with gradient operators [8]. A pair of Sobel masks H_1 and H_2 measure the gradient g of the image u in two orthogonal directions:

$$H_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, H_2 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix},$$

$$g = \sqrt{(u * H_1)^2 + (u * H_2)^2}.$$

A threshold is computed using the cumulative histogram of g, so that 10% of pixels with largest gradients are declared edges. In a "mixed" algorithm, range blocks containing at least one pixel declared edge is coded with the L_1 norm. The L_2 norm is used for other blocks. For 256×256 images and 8×8 range blocks, only one third of the range blocks are coded with the L_1 norm, with a corresponding reduction in computation time.

4 RESULTS

4.1 Signal-to-Noise Ratio

The general definition of the signal to noise ratio (SNR) is:

$$\text{SNR} = 20 \log_{10} \frac{\text{(original energy)}}{\text{(error energy)}}$$

The energy of a signal is its norm, leading to two different definitions:

$$\text{SNR}_{L_2} = 10 \log_{10} \frac{\sum_{i=0}^{M^2-1} (\text{original}_i)^2}{\sum_{i=0}^{M^2-1} (\text{original}_i - \text{coded}_i)^2}$$

$$\text{SNR}_{L_1} = 20 \log_{10} \frac{\sum_{i=0}^{M^2-1} |\text{original}_i|}{\sum_{i=0}^{M^2-1} |\text{original}_i - \text{coded}_i|}$$

where original_i and coded_i denote pixels in the original and coded images, respectively. As expected, the signal-to-noise ratio computed with SNR_{L_1} is always better with the L_1 norm than with the L_2 norm, and vice-versa. Hence, no conclusion can be drawn from the numerical results.

4.2 Subjective Visual Quality

The L_1, L_2, and mixed algorithms are qualitatively evaluated from the decoded images. Because of the large computational burden, even with the mixed algorithm, the algorithms are tested only with 256×256 images and 8×8 range blocks. Since the blocks are relatively large compared to the image size, none of the coded images are of very high quality. However, some promising conclusions can be drawn.

Edges are clearly better coded and images are globally sharper with the L_1 algorithm. There are no visual differences between the L_1 algorithm and the mixed one. Uniform areas are similarly coded with the three algorithms.

Figure 1 presents the original 256×256 "camera" image. Figures 2-4 are the decoded images obtained with the three algorithms for 8×8 range blocks and a bit rate of 0.27 bit per pixel. The quantization of the transformation offset coefficients, t, is based on a Laplacian density, while the scaling coefficients, α are quantized based on a uniform density. With the L_1 and mixed algorithms, edges are sharper, see the man's eye. But some other details disappear, see the camera in the man's hand. Figure 5 is the original 256×256 "ape" image, and figures 6-8 are the decoded images.

5 CONCLUSION

A new approach for a better coding of edges was presented here. The L_1 norm was used instead of the L_2 norm. It implies a considerable increase of the computational load, reduced by including an edge detection pre-processing.

Further improvements include a better classification of edge/non-edge range blocks. For example, textures are detected as edges, although they are not better coded with the L_1 algorithm. More efficient edge detection procedures and solutions of the non-linear minimization problem could also be used.

Figure 1: Original 256×256 "Camera" Image.

Figure 2: "Camera," L_2 Algorithm: $\text{PSNR}_{L_2} = 22.48$ dB, $\text{PSNR}_{L_1} = 28.19$ dB, bit rate = 0.27 bit per pixel.

References

[1] A. E. Jacquin, *A fractal theory of iterated Markov operators with applications to digital image coding.* PhD thesis, Georgia Institute of Technology, 1989.

[2] E. Yuval Fisher, *Fractal Image Compression: Theory and Application.* Springer-Verlag, 1994.

[3] R. A. DeVore, B. Jawerth, and B. J. Lucier, "Image compression through wavelet transform coding," *IEEE Trans. Inform. Theory*, vol. 38, pp. 719–746, Mar. 1992.

[4] B. Girod, "What's wrong with mean-squared error?," *Digital Images and Human Vision, edited by A. B. Watson, MIT Press*, vol. Chapter 15, pp. 207–220, 1993.

[5] I. Barrodale and F. D. K. Roberts, "An improved algorithm for discrete L1 linear approximation," *SIAM Journal Numerical Analysis*, vol. 10, pp. 839–848, Oct. 1973.

[6] S. I. Gass, *Linear Programming: Method and Applications, 5th Ed.* New-York: Mc Graw Hill Book Co., 1985.

[7] I. Barrodale and F. D. K. Roberts, "An efficient algorithm for discrete L1 linear approximation with linear constraints," *SIAM Journal Numerical Analysis*, vol. 15, pp. 603–611, June 1978.

[8] A. K. Jain, *Fundamentals of Digital Image Processsing.* Prentice Hall, 1989.

Figure 3: "Camera," L_1 Algorithm: $\text{PSNR}_{L_2} = 21.98$ dB, $\text{PSNR}_{L_1} = 28.62$ dB, bit rate = 0.27 bit per pixel.

Figure 4: "Camera," Mixed Algorithm: $\text{PSNR}_{L_2} = 21.94$ dB, $\text{PSNR}_{L_1} = 28.62$ dB, bit rate = 0.27 bit per pixel.

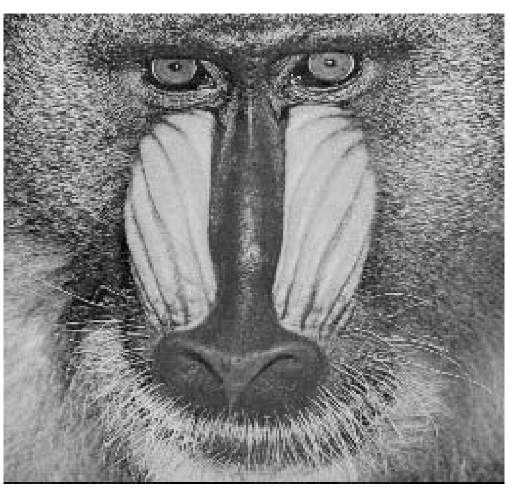

Figure 5: Original 256×256 "Ape" Image.

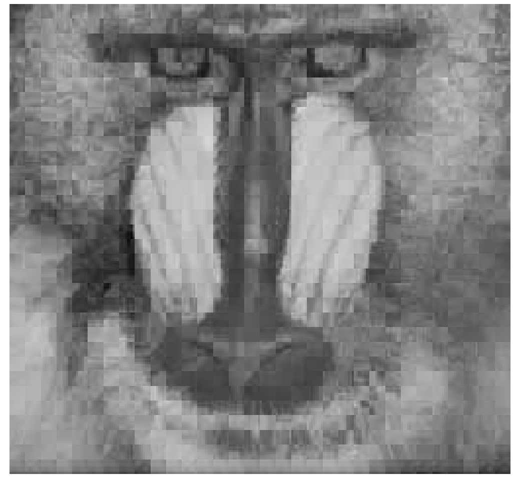

Figure 6: "Ape," L_2 Algorithm: $\text{PSNR}_{L_2} = 18.95$ dB, $\text{PSNR}_{L_1} = 21.61$ dB, bit rate = 0.27 bit per pixel.

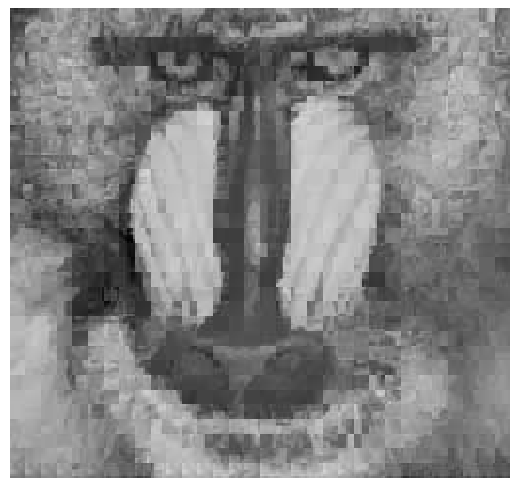

Figure 7: "Ape," L_1 Algorithm: $\text{PSNR}_{L_2} = 18.50$ dB, $\text{PSNR}_{L_1} = 21.63$ dB, bit rate = 0.27 bit per pixel.

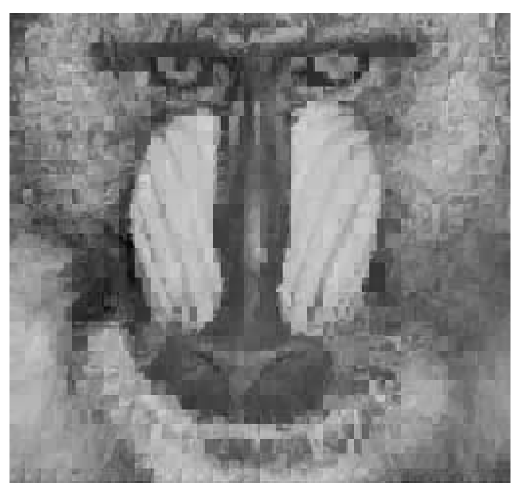

Figure 8: "Ape," Mixed Algorithm: $\text{PSNR}_{L_2} = 18.53$ dB, $\text{PSNR}_{L_1} = 21.63$ dB, bit rate = 0.27 bit per pixel.