Recursive Histograms Comparison for Accelerating Fractal Image Compression

Eric Amram

ENSTA
32, bd Victor
75015 Paris, France
amram@ensta.fr

Jacques Blanc-Talon

CTME/GIP
16bis, av. Prieur de la Côte d'Or
94114, Arcueil, France
blanc@etca.fr

Abstract

A new recursive algorithm based on Jacquin's general scheme is proposed in which a predicate is used for checking whether a domain block may match a given range block. Histograms comparison is used and dramatic accelerations of the encoding process have been measured. This technique is totally compatible with other acceleration techniques. Results on generic images are shown.

1 Introduction

Whatever the technological improvements in terms of memory size and transmission time will be, it is not risky to say that compression tools will always be if not necessary, at least useful.

Storing images in less memory cuts storage cost, and transmitting compressed codes instead of full video frames allows transmitting more. Numerous techniques competing in this technological race may be classified according to a simple criterion. They may be lossless (for instance, Huffman's encoding and run length encodings, Lempel/Ziv coding are lossless techniques), in which case it is still possible to compute the initial image from the compressed one; they may be lossy (DCT, wavelet transform, vector quantisation, spline-based methods, cellular automata and fractal techniques: IFS, PIFS, RIFS, IFSM, etc, are lossy) and then the original image is lost, replaced by an approximate one. Lossless techniques have to be preferred when acccuracy is needed but their compression ratio is theoretically bounded and remains generally quite small. On the converse, lossy techniques may reach any ratio provided that the human eye is misled by the approximate images they yield.

Before putting an algorithm in the public domain, a couple of key issues have to be won. The first one aims at achieving the highest possible compression ratio (in fact, a higher one than any other of any existing algorithm) while keeping the best possible quality of the image after decompression. The second one aims at encoding any image in a reasonable time. Thus, the main part of the research effort concerns the encoding step whose running time and quality determine the validity of the whole technique: the decoding process consists most of the time in iterating a simple algorithm.

Fractal Image Compression, as introduced by M. Barnsley *et al* [1], makes use of some basic properties of fractal geometry which forecast better ratios than any other technique. Today, the performances are promising but still have to be improved. About the compression time, it seems rather difficult in the case of Jacquin's algorithm[6] to prune the possibilities tree significantly even if recent advances have been made[9, 14]. About the compression ratio, one has to admit that the block-coding compression scheme does not take into account any geometrical property of the original image at all. Modifications either of the transform maps [11] or of the initial scheme [2, 3, 10, 15, 16] have been proposed in order to improve (significantly) the quality of the images. Let us mention a couple of pessimistic studies [5, 13] which help in keeping enthusiastic papers at a healthier level!

There are two types of accelerating techniques: those which are lossless, i.e. which do not sacrifice any image quality for the sake of the speedup, as in [4, 9, 14]; those which are not, such as [7, 12] or the present one. Since fractal compression is already a lossy technique, it is not very grave, after all, to lightly alter the final image quality!

According to Jacquin's scheme, the image is divided into range blocks. For each range block, we find in the image a domain block twice as large that best matches the range block after size reduction (averaging), isom-

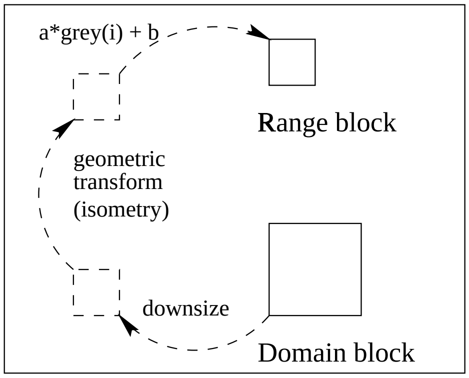

Figure 1: From domain to range block

etry and luminance affine transformation (see fig. 1). The Contraction Mapping and Collage theorems state that applying these transforms to any image will lead to the IFS attractor, which is close to the image in the Hausdorff metric. Thus, the IFS code *is* the image.

In our implementation, the search is made on the whole reduced image and not only around the range block. We use non-overlapping domain blocks, but nothing forbids doing the opposite. The decoding process is a simple sequence of iterations: starting from any image, the previous transformations are applied until convergence is reached, which is fast and easy.

2 Quick-Search Algorithm

Let us consider the whole reduced image and and ask:

Can the range block belong to it ?

If yes, we split the box into four parts, each one corresponding to the possible top left corner positions of the range block (see figure 2).

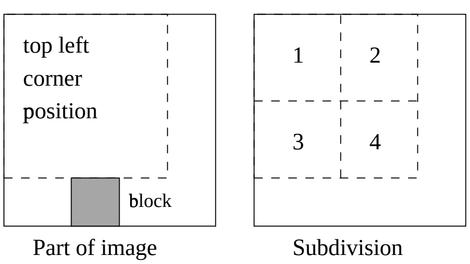

Figure 2: Histogram quadtree

Anmy onotnousnftuinof rom a set (the s
gryealues) hottcharset winhfaorredleart hi

it ($\mathbb{R}$ fitted with $\leq$ for instance) cou
predicate. We use here a phairsitsognram-
scheme that pwaeiwibldleoxw. The initia
plexity which was $O(X \times Y$ becomes O
comparisons ($X = imagewidth/bloc$
comparisons are made on histogram
found, which meaarnse thalte pfthnyeent o
sizfetohe obkls.

Consideringf tahpe aimtage, we build it
tografgmroy values. We deffsiqneaarneu mbe
(we otk for the sfaskemplicity) and we di
the rafnggreeoy valfuoemnlyEach square re-
flects the nfupmixbeelrsow ithin the interva

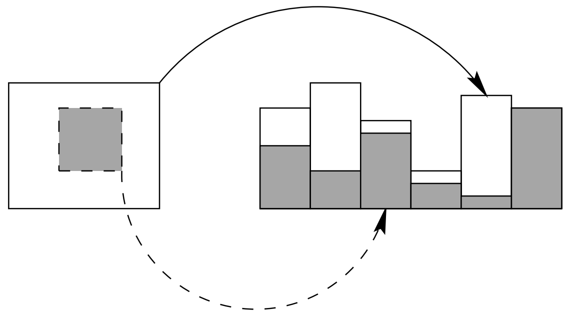

Figure 3: Blocks and histograms inclusion

As we allow the luminance of the block to be transformed affinely, we have to normalize the grey values of a block before we compute its histogram.

The predicate is thus the following: a range block belongs to a part of the reduced image if its histogram is "included" in the histogram of the part. In this case we subdivide the search space of the top left corner. This is of course valid since we consider only Jacquin's isometries to transform the block: there is neither filtering nor approximation on the grey values between range and domain blocks once the downsizing of the image has been made.

In practice we first precompute the quadtree of domain blocks histograms, for a given block size: the root is the entire image, whereas the leaves are the actual domain blocks. The histogram of each range block is computed in the main loop and we compare it recursively to the nodes of the domain blocks quadtree. This yields a few possible solutions and we evaluate their scores by the usual rms method. The Quick-Search algorithm being compatible with [8] and [14], we can expect very good encoding time with a combined approach.

Please note that there is nothing to prevent our using overlapping domain blocks. This option is currently not included in our results, but we envision to add it to improve the image quality.

The method would be totally lossless if the matches were almost perfect. In practice only very small squares (3×3) respect this condition, imperfection being the rule for bigger ones: we have to minimize errors (rms for instance).

That is why we have to introduce some flexibility in histograms comparisons. Among all possibilities, overlapping intervals seem to be the most efficient. If we allow too much flexibility, we have indeed to face a lot of useless direct comparisons between blocks, as in the full search process.

What we call "overlapping intervals" are special histograms for domain blocks only. If a grey value is near another square, then it belongs to both of them.

3 Results

All computations were done on a hyper-sparc Sunstation with a single processor. Results express the CPU time needed to achieve the compression of the image.

In this version, domain blocks are not overlapping so the quality of the decoded images is not the highest we can expect, as already mentioned. Our only purpose here is to show the speed gain factor between the full search method and the Quick-Search algorithm. The full search program has been fully optimized to measure the real improvement.

Good matches remain as good as they were. However bad matches tend to become worse : this explains the difference occurring with large range blocks between the SNR of the full search and the SNR of the Quick-Search Algorithm. A 3×3 filter (weighed average) has been applied after decompression to make blockiness artifacts disappear (when $blocksize > 4$).

CPU time was measured with the Unix time command. See table 1.

4 Conclusion

This new technique has shown sharp cuts in the encoding time, without significant loss of the global image quality. With an average speedup factor of 8, we are convinced that this kind of techniques could help a lot in putting compression times at a realistic level. We are currently focusing on the definition of new predicates as well as the algebras they generate.

Attempts for parallelizing the encoding process have shown that it could take less than one minute for processing very *big* images. The combination of

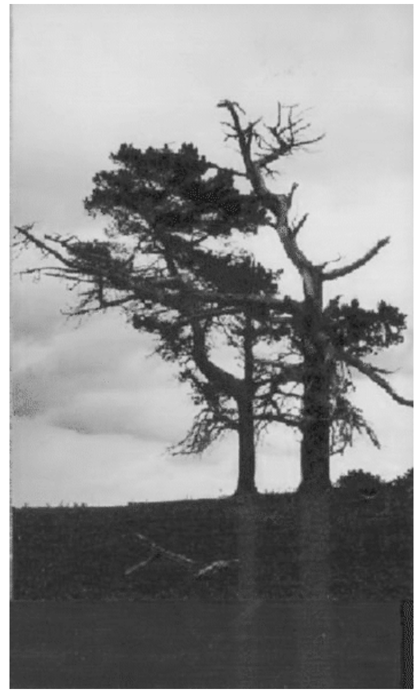

Figure 4: Australian trees (353×599)

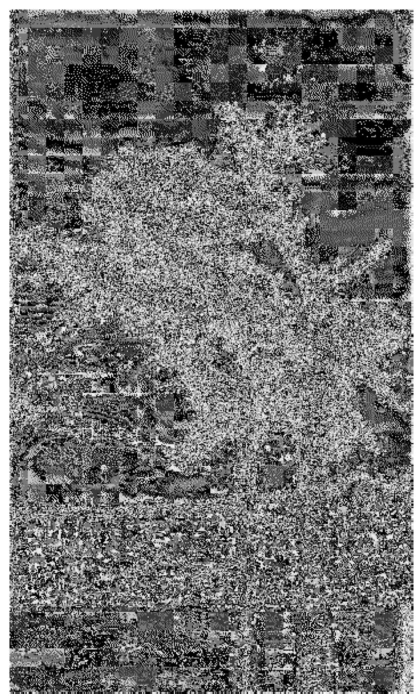

Figure 5: Normalized error image: max error = 59, 16×16 blocks, PSNR = 33.9 dB

our algorithm with other acceleration methods seems very promising for personal computer applications.

Acknowledgments

Thanks to S. Teller (M.I.T.) and M. Werman (Hebrew University) who have had the seminal idea of histograms comparison.

Table 1: CPU encoding time and PSNR

Block size	**Australian trees** 353×599		
	Quick-Search	Full search	Speedup
4×4	31min20s 38.35 dB	118min 38.35 dB	**4.1**
8×8	5min59ss 37.5 dB	31min 28s 37.9 dB	**5.7**
16×16	47s 32.6 dB	7min 3s 33.9 dB	**9**
Block size	**Lena** 512×512		
	Quick-Search	Full search	Speedup
4×4	36min 19s 34.5 dB	261min 34.5 dB	**7.2**
8×8	13min 16s 28.6 dB	102min 23s 29.5 dB	**7.7**
16×16	2min 04s 23.2 dB	18min 46s 24.6 dB	**9.1**

References

[1] M. Barnsley. *Fractals everywhere.* Academic Press, 1988.

[2] G. Cretin. Fractal image compression: experiments on HV partitioning and linear combination of domains. In Jacques Lévy-Véhel, Evelyne Lutton, and Claude Tricot, editors, *Fractals in Engineering*, Arcachon, France, 1997. INRIA, Springer-Verlag.

[3] F. Davoine, G. Robert, and J.-M. Chassery. How to improve pixel-based fractal image coding with adaptive partitions? In Jacques Lévy-Véhel, Evelyne Lutton, and Claude Tricot, editors, *Fractals in Engineering*, Arcachon, France, 1997. INRIA, Springer-Verlag.

[4] Kenneth M. Dawson-Howe. Lossless image compression using a simple prediction method. Technical report, Trinity College, 1996.

[5] F.M. Dekking. Fractal image coding: some mathematical remarks on its limits and its prospects. Technical Report 95-95, TU Delft, 1995.

[6] Arnaud E. Jacquin. A novel fractal block-coding technique for digital images. In *Int. Conference on Acoustics, Speech and Signal Processing*, pages 2225–2228. IEEE, 1989.

[7] H. Lin and A.N. Venetsanopoulos. Fast pyramidal search for perceptually based fractal image compression. In *ICIP*, volume 3, pages 596–599, Washington, USA, October 1995.

[8] H. Lin and A.N. Venetsanopoulos. A pyramid algorithm for fast fractal image compression. In *IEEE International Conference on Image Processing*, volume 3, pages 596–599, Washington, USA, October 1995.

[9] David John Nettleton and Roberto Garigliano. Reductions in the search space for deriving a fractal set of an arbitrary shape. *Jour. of Mathematical Imaging and Vision*, 1996.

[10] Dan Popescu and Don Bone. Permutation based fractal image coding. In Jacques Lévy-Véhel, Evelyne Lutton, and Claude Tricot, editors, *Fractals in Engineering*, Arcachon, France, 1997. INRIA, Springer-Verlag.

[11] Dan Popescu, Alex Dimca, and Hong Yan. Generalized square isometries - an improvement for fractal image compression. In *8th Int. Image Conf. on Image Analysis and Processing*, San Remo (It), 1995.

[12] Dietmar Saupe. Accelerating fractal image compression by multi-dimensional nearest neighbor search. In *Data Compression Conference*, Freiburg, Germany, March 1995. IEEE.

[13] Dietmar Saupe. The futility of square isometries in fractal image compression. In *International Conference on Image Processing*, Lausanne, September 1996.

[14] Dietmar Saupe and Hannes Hartenstein. Lossless acceleration of fractal image compression by fast convolution. In *International Conference on Image Processing*, Lausanne, September 1996.

[15] Masayuki Tanimoto, Hiroshi Ohyama, and Tadahiko Kimoto. A new fractal image coding scheme employing blocks of variable shapes. In *International Conference on Image Processing*, Lausanne, September 1996.

[16] N. Wadströmer. Coding binary fractal images with partitioned iterated function. In Jacques Lévy-Véhel, Evelyne Lutton, and Claude Tricot, editors, *Fractals in Engineering*, Arcachon, France, 1997. INRIA, Springer-Verlag.