

On Fractal Compression

Bogdan Nedelcu

"Politehnica" University of Bucharest
313 Splaiul Independentei,
77206 Bucharest
ROMANIA
E-mail: bogdan_nedelcu@hotmail.com

Abstract: A great number of procedures have been and are used in the image compression field. Most of them have, inspired by standard data compression algorithms and data signal processing, reached a certain limit and become obsolete. This paper attempts to emphasize a new way of compression, somehow different from the current procedures, which yields surprisingly for a specific set of images – the nature-generated ones.

Keywords: fractals, compression

1. Introduction

Fractal compression is one of the most exotic procedures in image data compression known so far. As nature-generated images contain a lot of pieces similar to each other, researchers cannot help being surprised with such a method, more promising than any other and distinguishable from all the others in that the image is not viewed as a signal, or as a succession of numbers, but as a picture. The whole process of image compression minds the fact that images are nature-generated and that the human eye may not perceive the details possibly lost during this type of image codification process.

Unlike other image compression techniques, this one does not need a standard algorithm, but, some major steps are to be taken, so that a particular method is called a *fractal compression algorithm*. Here are described only black and white images, as the case of colour images is just a simple application of the gray scale algorithm.

2. Image Coding

A digital image can be stored in a matrix such that every matrix element represents the gray-level of a corresponding image pixel. For simplicity, let us suppose the matrix being squared – $n \times n$. To actually compress such an image (matrix), the finding of another representation of the matrix is intended, a

function-like representation, which will need less space to be stored in.

Collage Theorem

Given $F: M \rightarrow M$ a contraction, and $g \in M$, g^* being the fixed point of F ,

$$\text{if } \|F(g) - g\| < \varepsilon \Rightarrow \|g^* - g\| < \frac{\varepsilon}{1 - s}$$

$$\text{where } s = \sup_{h \in M_n(\mathbb{R})} \frac{\|F(h)\|}{\|h\|}$$

In other words, the more the image transformed through the function will match itself, the more the fixed point of the function will match the image.

Jacquin and Barnsley have observed that certain classes of contractions have very interesting fixed points: natural images. The general form of these functions is: $f(x) = s \cdot T(x) + c$; $s \in (-1, 1)$ and $c \in (0, 255)$ where s is a contrast factor, c is a brightness adjustment and T -an affine transformation. The fractal compression algorithm tries to find a function whose fixed point matches almost perfectly the given image. To find such a function, by Collage Theorem, is to find F such that F renders an image closer to itself.

Step 1: split the matrix (call it A) into N non-overlapping partitions, of size $B \times B$, and call them *range cells*, $\{R_i\}$; $i=1, N$. Split again matrix A into M parts, of size $2B \times 2B$ and call them *domain cells*, $\{D_i\}$; $i=1, M$.

Step 2: for each range block R_i try to find through the domain blocks a certain D_j which, if transformed by (1), is the best approximation that could be found, of R_i

$$D_i' = s_i \cdot T(D_i) + c_i \quad (1)$$

where T is an affine transformation (rotation, mirroring, etc.) followed by a mapping from a

2Bx2B matrix onto a BxB one. D_i' will be the D_i matrix seen at a different scale after having suffered a certain size transformation. If a right match cannot be found, the range block will be split into four equal pieces, each one being a $B/2 \times B/2$ matrix and the search for a good approximation of each of the four new matrices will go on. There will certainly be an end to this process as, when we get an 1x1 matrix, we can choose c_i as equal to the gray level of the matrix and s_i as 0, making no assumptions about the value of D_i .

Denoting μ_{R_i} a matrix having 0 outside the region corresponding to the range block R_i and the values of R_i inside, we can write:

$$F = \sum_{i=1}^N \mu_{D_i} \quad (2)$$

If $s_i < 1$ for any $i=1, N$ then it can easily be proven that F is a contraction. Now, it is obvious that the algorithm will find out a function attempting to convert a given matrix to itself – so it has the fixed point quite close to the original image. Because this function can be stored in less space than matrix A (we have only to remember s_i, c_i , the positions of D_i and the corresponding affine transformation), we use compression.

3. Image Decoding

The image decoding process is a simple iteration of the identified function F , starting at any initial point – say an empty matrix. After a certain number of iterations, the fixed point is reached and the image is fully reconstructed.

4. Strengths and Weaknesses

This method's main advantage over the usual image compression techniques is its high rate of compression obtainable in the case of natural images.

The main problem with the algorithm is the process of searching for good pairs of range and domain blocks. If the set of domain blocks is limited, we miss a good approximation of a certain range block, and the image will lose its quality characteristics. If the set is too large, the construction time of function F will be unacceptable. Also, the more affine the

transformation used, the better the images obtained, but a longer time will be taken by the compression process.

5. Experimental Results

The goal of the experiments was to reach a better compression rate than the general algorithm usually does. While experimenting, on a set of different images, various modifications of the parameters of the codification algorithm, an increase of the compression factor has been observed, when range blocks of a bigger size (16x16 or 32x32) had a match in the domain set.

Transformations having a fixed point such as blocks (images), are difficult to find, and usually the approximations made instead yield a poor quality of the image, many of the details being lost. For finding such transformations, searching for a good match all through the domain and range sets, is a very time-consuming task, so it could actually take several minutes for an image, size 256x256, to be compressed at a reasonable compression rate.

It is faster machines which can help overcome searching time problems. Finally the following situation is reached: matches between big range blocks and domain blocks lack in the current image. As the splitting of the block would take us back to the standard algorithm, the author has chosen to introduce more affine transformations. The algorithm, over its main part, tries to find some similarities between blocks of the image. Using conventional affine transformations like mirroring, rotation or symmetry, the process of finding big range blocks would have had very few chances of success. So some "wired" transformations were introduced: shearing composed with rotations, symmetry with respect to some other points than the centre of the image in the plane, bit-rotation, etc.

These new affine transformations proved to be an excellent solution to the quality of the compressed image. The remaining problem was in the amount of coded data becoming larger, as more and more numbers representing the newly introduced transformations needed be stored.

Keeping in mind that the algorithm uses these "wired" transformations only for some isolated cases of blocks, the author actually compressed them with a standard compression algorithm and obtained some interesting results. By searching mainly for 16x16 and 32x32 domains, and using a simple Huffmann compression algorithm for the transformation set, the compression factor goes from 18:1 to 20:1 or 22:1 and even higher, depending on the image being used. Such a performance can also be noticed in the quality of the image, not in the first reconstructed image, but in the one zoomed by 2 or 4. The details are better in zooming when bigger blocks are being used.

As shown, in the codification algorithm the result (the compressed image) has no element which depends on the size of the original image. In the standard compression methods, the image is codified with respect to the colour or the gray level of the points it contains. Here results a set of relations between parts of the image itself. So the reconstruction is no longer dependent on the original picture size. One most interesting fact related with this compression method is the following: images can be reconstructed at a different scale, and still keep the aspects that would have, by usual zooming, been lost in the original ones. An image zoomed eight times becomes a picture of big squares – representing the pixels of the image being processed. To eight -time zoom a fractal compressed image is to get a picture which still preserves its curves, contours and shade parts – which are very important details to the human eye.

6. Conclusion

Fractal image coding is an interesting approach to image compression, which concerns the data to be actually compressed as an image, not as a succession of bits, or as a signal.

By adding more and more affine transformations, the probability of finding relations between different parts of the image has increased. The set of new transformations proposed is no longer limited as the set of range and domain blocks did. So the search algorithm for relations between parts of the image will have no longer to split the range blocks for which a good match could not be found—instead it will just use one of the new special transformations introduced. An interesting aspect would be the construction of these "wired" transformations from the image itself, somehow to deduce them from the characteristics of the original picture.

This would eventually reduce the codification time and would also keep the compression factor at higher levels than standard compression methods usually do.

REFERENCES

- BEDFORD, T., DEKKING, F.M. and KEANE, M.S., **Fractal Image Coding Techniques and Contraction Operators**.
- JACQUIN, A. E., **Fractal Image Coding: A Review**, Proceedings of the IEEE, Vol. 81, No. 10, October 1993.