Local Search Fractal Image Compression for Fast Integrated Implementation

Nguyen T. Thao
Dept. of EEE, The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong, tel/fax: 852-2358-7063/1485, eethao@ee.ust.hk

Abstract— **The well known drawback of fractal image coding is its heavy computation complexity. In this paper, we show however that only a very small portion of this complexity is needed to encode the contours and the smooth areas of the objects of a certain minimum size. This is simply achieved by a local search of the matching domain blocks without isometry. Therefore, instead of forcing the fractal coder to take care of all types of feature in the image, we limit itself to encoding the features of the type mentioned above. We give the role of capturing the missing detailed features to a DCT coder. This is much more cost effective than using a complete self-sufficient fractal coder. We provide a special method of fractal-DCT output recombination which prevents the introduction of the classic DCT artifacts while preserving the qualities of the local search fractal information. We give experimental results at the given total bit rate of 0.18 bit/pixel.**

I. Introduction

For the need of multimedia, three goals must be achieved simultaneously when developing a new image compression scheme: (1) good image quality; (2) low bit rate; (3) feasibility of integrated implementation. Substantial research has been devoted to the method of fractal image compression [1], [2]. While this technique has good potential for the criteria (1) and (2), it is particularly weak for the goal (3). Indeed, it is known that a heavy computation complexity is required for the search of matching domain blocks in the encoder. In this paper, we show that a very large portion of this complexity is in fact used for the encoding of the detailed features of the image only. Indeed, when confining the matching block search to a region local to every given range block and dropping all isometry comparisons, we show that an image with very good contours and smooth areas can be reconstructed while the number of block comparisons par range block can be reduced from 2×10^6 to 81 for a 512×512 image and an 8×8 range block partitioning. Moreover, we show that the local matching block search permits a new type of lossy and lossless compression of the fractal parameters, thus showing an extra contribution of this version of fractal coding with respect to the goal (2). This shows that fractal coding is very efficient at encoding the edges and the smooth areas, while requiring an extremely high marginal cost of computation when trying to encode all features of the image. The basic idea of this paper is to limit the use of fractal coding to its efficient mode, that is, to only encoding the edges and the smooth parts of the image, and give the role of encoding the missing detailed features to another coding scheme which is more effective. In this paper, we use DCT coding for dealing with the detailed features. For this purpose, a special image recombination method is used, based on projections onto convex sets (POCS) [3], [4], to prevent the introduction of the DCT artifacts. Finally, we show experimental results at the bit rate of 0.18 bits/pixel.

II. Review on fractal coding

The basic version of fractal coding consists of splitting the image into a regular partition of 8×8 blocks B_i (like DCT coding) called the range blocks (see Figure 1). Then for each block B_i, the fractal encoder searches for the contrast coefficient s_i, the DC offset d_i, the 16×16 block D_i in the image and the isometric transformation $\mathcal{T}_i$ such that the transformed block

$$s_i\cdot\mathcal{T}_i\left(\mathcal{S}(D_i)\right)+d_i$$

is as close as possible to the block B_i in the mean squared error sense ($\mathcal{S}$ is a decimation operator, "shrinking" a 16×16 block into an 8×8 block). Then, the coefficients s_i and d_i, the location of the block D_i and the choice of isometry $\mathcal{T}_i$ are encoded and sent to the decoder.

The decoder reconstructs an approximation of the original image, called the fractal image, by iteratively forcing each range block B_i of an initial image estimate to be equal to $s_i\cdot\mathcal{T}_i\left(\mathcal{S}(D_i)\right)+d_i$.

The heavy computation complexity is due to the fact that the choice of block D_i and isometry $\mathcal{T}_i$ is obtained by exhaustive search in the full image. Typically, for a 512×512 image, with 8 choices of isometry, this requires approximately 2×10^6 block comparisons for each range block B_i. Moreover, in order to well reproduce the details within the 8×8 blocks, a more sophisticated version of fractal coding has to be introduced, where the particularly detailed blocks are split into subblocks on which fractal coding is applied (quadtree partitioning). This technique adds to the difficulty of integrated implementation by making the algorithm less systematic and irregular.

III. Local fractal coding

We propose to test a very reduced version of fractal coding where

(1) the range block partitioning is reduced to the 8×8 regular block size,
(2) the search for a matching block is local and limited to the domain blocks which contain the considered range block (see Figure 1),
(3) no isometry is considered in the block comparisons.

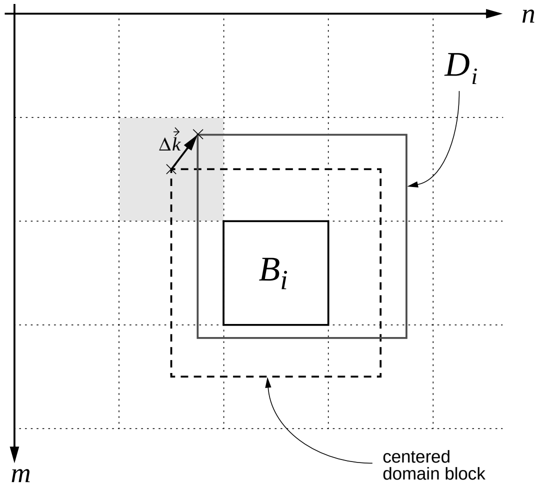

Fig. 1. Range block partitioning and local domain blocks.

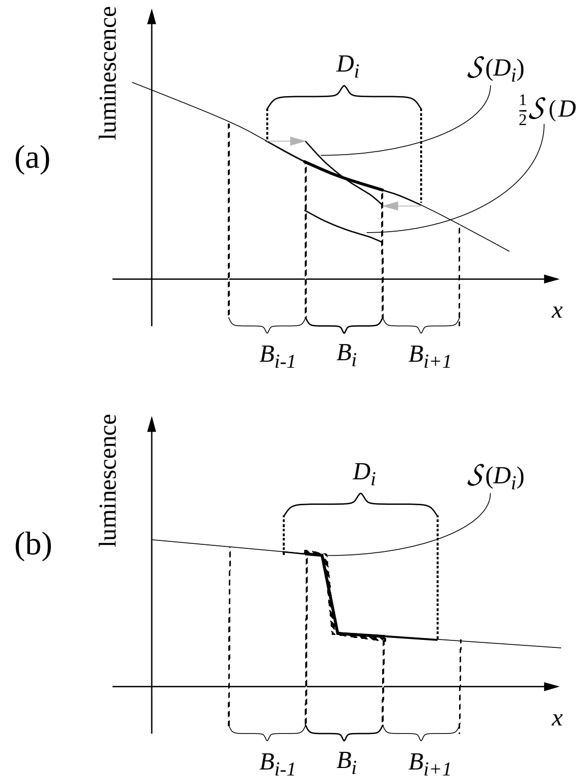

Fig. 2. One dimensional illustration of fractal search of the domain block D_i for a given range block B_i in the case of smooth variations (a) and an edge (b). The figures show the choice of domain block D_i local to B_i such that $\frac{1}{2}\mathcal{S}(D_i)$ approximates B_i up to some DC offset in the case of (a), and $\mathcal{S}(D_i)$ approximates B_i in the case of (b).

We thus reduce the number of block comparisons per range block from 2×10^6 to 81. We apply this coding scheme to the image of Figure 3 and show the resulting fractal image in Figure 4(a). In spite of the drastic complexity reduction, one can see that the objects larger than the 8×8 block size in the decoded image have contours and smooth areas of particularly high quality (observe for example the smoothness of the shoulder without blocking artifacts, in contrast with the sharp definition of its contour). This result is in fact not so surprising. The intuition that a local search fractal encoder can easily deal with either the smooth areas or the contours of objects larger than the range block size is graphically shown in one dimension in Figure 2.

IV. SPECIAL FRACTAL PARAMETER BIT RATE COMPRESSION

Another contribution of the local search is the output bit rate reduction. We have indeed reduced the bit rate for the characterization of the isometry from 3 bits/block to 0 bit/block and the bit rate for the matching block position from 18 bits/block (9 per dimension) to 6.3 bits/block using Huffman coding. Now, in the following paragraphs of this section, we show that the specific context of local search permits an extra and special compression of the fractal parameters.

A. Compression of s_i

Figure 2 shows that the values 1/2 and 1 are typically sufficient discrete values for s_i to perform a satisfactory block matching for smooth areas and edges respectively. From this initial intuition, we limit s_i from the beginning to the values 0, 1/2, 1 when finding the best approximation of B_i by $s_i \cdot \mathcal{S}(D_i) + d_i$. The value 0 permits the approximation of B_i by a simple DC component. By using Huffman coding, we obtain a bit rate for s_i of 1.25 bits/block.

B. Compression of d_i

Not much image degradation is induced when quantizing the coefficients d_i with a step size of 4 for a maximum pixel value of 256. Using Huffman coding, we obtain a bit rate of 4.9 bits/block for d_i. In fact, further compression can be achieved thanks to the local search condition. In the matching process, it is easy to show [2, p.21] that the parameter d_i should be chosen such that $\overline{B_i} = s_i \cdot \overline{D_i} + d_i$, where $\overline{B_i}$ and $\overline{D_i}$ are the DC components of B_i and D_i respectively. Now, in our special version of fractal coding, D_i is local to B_i. If we write $\overline{D_i} = \overline{B_i} + \epsilon_i$, we therefore expect ϵ_i to be relatively small. This implies that

$$\frac{d_i}{1-s_i} = \overline{B_i} + \frac{\epsilon_i}{1-s_i} \simeq \overline{B_i}.$$

While little lossless compression can be achieved on the direct coefficient d_i, spatial correlation can be exploited from the coefficient $\overline{B_i}$. When encoding $d_i/(1-s_i)$ in a way similar to DC components in JPEG, we reduce the bit rate for d_i from 4.9 bits/block to 3.9 bits/block.[1]

C. Further lossy compression and result

With some controlled loss of image quality, we can further reduce the total bit rate. When encoding the position of the matching block as a displacement vector (see Figure 1), the vector appears to have a quite uniform distribution. This prevents Huffman coding from yielding good data compression. In fact, for a smooth area, it is easy to see that the domain block centered around the range block can be typically taken as a satisfactory matching block, although it may not be absolutely the best one in the MSE sense. In [5], we propose a method to increase the probability of appearance of certain "prioritized" displacement vector locations while inducing a very limited loss of image quality. Using again Huffman coding, we reduce the bit rate from 4.1 bits/block to 2.4 bits/block.

Using a similar approach, we also further reduce the bit rate for s_i from 1.25 bits/block to 0.6 bit/block by increasing the occurrence of the discrete value of 1/2 and performing a runlength coding on this value.

As a result, we obtain the fractal image of Figure 4(b). In spite of all the above lossy compression operations, the image degradation with respect to Figure 4(a) is in fact quite limited and the features we are mainly interested in (contours and smooth areas) are well preserved. The total bit rate is 0.11 bits/pixel.

V. COMPLEMENTARY DCT ENCODING AND RECOMBINATION

To encode the missing information of the details smaller than the range block size, we start proceeding like in pyramidal coding. We include the fractal decoder at the encoder side. Thus, the difference between the fractal decoded image and the original image can be measured and encoded through DCT. However, to avoid introducing DCT artifacts, we recombine the fractal and DCT output codes in a way different from pyramidal coding in general. It was shown in [4] that the complete information provided by a DCT code is not a single approximation of the original image, but the definition of a whole set of possible images that the original image could deterministically be. The classical DCT decoded image was shown to be a particular choice of element in this set, and precisely, its center. To avoid integrating the artifacts generated by the classically DCT decoded image, we choose another element of the set obtained by POCS [3], [4]. More details are given in [5]. Also, as the fractal image already contains high quality information on the block DCT components, the DCT coefficients corresponding to these components are simply dropped.

We show in Figure 4(c) the recombined fractal-DCT image. One can see that all the features already well reproduced by the pure fractal image of Figure 4(b) are preserved without any degradation. At the same time, a certain amount of details have been recovered. Note for example the recovery of the double edge of the brim of the hat.


Fig. 3. Original 512 × 512 image.

The bit rate yielded by the DCT encoder is 0.07 bit/pixel. This leads to the total bit rate of 0.18 bit/pixel. Although the PSNR is relatively low (28.8dB), we obtain a well balanced image in terms of perceptual features with no blocking or ringing artifacts (which is not the case for a purely DCT decoded image at this bit rate).

Figure 4(d) shows the result of a slightly modified decoding scheme, where an adaptive DCT domain weighted average is taken between the decoded image of Figure 4(c) and a purely DCT decoded image. While some DCT artifacts start to appear (artifacts on the contour of the left cheek), the global picture looks sharper and more detailed (with an increased PSNR of 29.1dB).

VI. FINAL REMARKS

Compared to the use of fractal coding alone, our method requires the extra use of a DCT encoder, a DCT decoder, a fractal decoder and a POCS (which amounts to a DCT encoder in terms of complexity). However, this added computation complexity remains negligible compared to the marginal cost of computation that would be needed for the local search fractal coder to be extended to a self-sufficient fractal coder. Our contribution is also more generally to motivate the method of hybriding several encoders of different types. This allows to optimize the global encoding scheme by splitting the image into several types of features and finding separately the most efficient encoder for each type of feature, in terms of simplicity of implementation, quality of feature extraction and bit rate compression altogether.

REFERENCES

[1] M.F.Barnsley and L.P.Hurd, *Fractal image compression.* Wellesley, Mass., 1993.

---

[1] The bit rate of 3.9 bits/block takes into account the fact that, when $s_i = 1$, we need to encode d_i separately.


(a) (b) (c) (d)

Fig. 4. Complete fractal coding process: (a) fractal image obtained from local domain block matching and no isometry; (b) fractal image with additional bit rate compression (0.11 bit/pixel); (c) fractal-DCT decoded image (0.18 bit/pixel); (d) modified fractal-DCT decoded image (0.18 bit/pixel).

[2] Y.Fisher, ed., *Fractal image compression : theory and application.* Springer-Verlag, 1995.

[3] R.Rosenholtz and A.Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding," *IEEE Trans. Circ. and Syst. for Video Tech.*, vol. 2, pp. 91–95, Mar. 1992.

[4] N.T.Thao, K.Asai, and M.Vetterli, "Set theoretic compression with an application to image coding," *Proc. IEEE Int. Conf. Image Proc.*, vol. II, pp. 336–340, Nov. 1994.

[5] N.T.Thao, "A hybrid fractal-dct coding scheme for image compression," *IEEE Trans. Circ. and Syst. for Video Tech.* To be submitted.