A HYBRID FRACTAL-DCT CODING SCHEME FOR IMAGE COMPRESSION

Nguyen T. Thao

Dept. of EEE, The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong

ABSTRACT

We introduce a new way to use fractal coding for image compression, based on the parallel use of a fractal encoder and a DCT encoder. The two encoders are given the complementary roles to capture the information of edge and smooth variation, and the information of detail respectively. We show the advantage of using this hybrid coding scheme over the use of a fractal encoder alone, or a DCT encoder alone. This coding scheme is also the occasion to demonstrate a new concept of coding by nonlinear feature separation based on regular and uniform algorithms, suitable for real-time VLSI implementation.

1. MOTIVATION AND OUTLINE

Substantial research has been done in image compression by the method of fractal coding [1, 2]. It is however known that in spite of its good potential for low bit rate compression, fractal coding implies a computationally very intensive encoding. Moreover, unlike DCT coding for example, fractal coding implies an inherently non-regular implementation requiring adaptive subblock partitioning. We briefly recall the principle of fractal coding in Section 2.

In this paper, we propose a new way to use fractal coding for image compression. To begin with, we point out that the major part of this computation complexity is required for the encoding of features smaller than the block size. Indeed, as shown in Section 3, a fractal encoder is already capable of well preserving the information of contour and smooth variation of objects larger than the block size, with a very reduced computation complexity. The basic idea of the paper is to use a fractal encoder only for the encoding of these contours and smooth variations. Because we know that using the fractal encoder to also capture the missing image details within the blocks would imply an excessive additional computation complexity, we give this task to a DCT encoder instead, working in parallel. DCT coding is classically good at capturing details within the block size while requiring a small computation complexity. We explain in Section 4 how the DCT encoder can be especially adapted in order to selectively capture the missing details. We also explain how the two parallel outputs of the fractal and DCT encoders can be recombined by the decoder in order to reconstruct a single image (method of convex projection or POCS). We give the result of an experiment at the bit rate of 0.18 bit/pixel. We end up with a hybrid fractal-DCT coding scheme with the following advantages:

(i) It relies on the fractal encoder for the contours and smooth areas to avoid the classical ringing and blocking artifacts of DCT coding.

(ii) It relies on the DCT encoder for the details within the blocks to avoid the excessive marginal computation complexity that the fractal encoder used alone would imply.

In Section 5, we explain that this coding scheme is in fact a preliminary demonstration for a new approach of image compression by feature extraction.

2. BACKGROUND 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).

3. PROPOSED FRACTAL ENCODER

Because we only give the fractal encoder the task to well reproduce the contours and smooth areas of objects larger than the 8×8 block size, we show in this section that we can drastically

(i) simplify the fractal encoder,

(ii) compress the fractal parameters.

Figure 1: Range block partitioning and local domain blocks.

Figure 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).

3.1. Local domain block matching

The fractal image shown in Figure 4(a) is obtained from the original image of Figure 3 in the case where no isometry is used and the search for a matching block. More precisely, we limit the search area to domain blocks which contain the considered range block as indicated in Figure 1. In spite of this drastic encoding simplification (the number of block comparisons per range block is reduced from 2×10^6 to 81 only), we see that the information of contour and smooth area still persists for objects larger than the range block size. This qualitatively show the intrinsic quality of fractal coding for this type of feature. With the local block search, we have also 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.

3.2. Compression of s_i

In the context of local matching, 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.

3.3. 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]

3.4. 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

---

[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.

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 [3], 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.

4. DCT PARALLEL ENCODING AND RECOMBINATION

To encode the missing information of the details smaller than the block size, we use in parallel a DCT encoder with the particular feature that the DC coefficients are dropped. The two fractal and DCT digital outputs are recombined at the decoder side according to the following procedure:

1 - Reconstruct the fractal image,
2 - Perform the projection (POCS) of the fractal decoded image onto the convex set associated with the DCT digital output according to the method of [4, 5].

The final decoded image is shown in Figure 4(c). 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.

To reduce the redundancy between the two parallel encoded outputs, we only send the difference between the DCT coefficients of the original image and the DCT coefficients of the fractal image. This of course requires that the fractal decoder be included in the encoder. With this method, the bit rate associated with the DCT coefficients in the experiment of Figure 4(c) 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).

Figure 3: Original 512 × 512 image.

5. FINAL REMARKS

The goal of this paper is not only to propose a new version of fractal coding but also to show some general advantages of hybriding several encoders of different types. This allows to optimize the global encoding scheme by 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. We have also demonstrated a method of image feature separation which

(i) is non-linear, unlike classical coding schemes based on linear decomposition (such as pyramidal coding and multiresolution analysis),
(ii) does not require image segmentation.

The aspect (i) is particularly interesting for very low bit rate image compression where the coding should be more oriented towards the extraction of perceptual features which are not fundamentally separable in a linear fashion. Concerning the aspect (ii), each of the involved parallel encoders have indeed a systematic and regular implementation, treating all parts of the image in a uniform manner. This is particularly needed for real time VLSI implementation.

The present experiment is a preliminary demonstration of these concepts.

6. REFERENCES

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

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

[3] N.T.Thao, "A Hybrid Fractal-DCT Coding Scheme for Image Compression", IEEE Trans. on Image Processing, To be submitted.

[4] R.Rosenholtz and A.Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding", IEEE Trans. on Circuits and Syst. for Video Tech., Mar. 1992.

[5] N.T.Thao and K.Asai and M.Vetterli", "Set Theoretic Compression With an Application to Image Coding", Proc. IEEE Int. Conf. Image Processing, vol. II, pp. 336-340, Nov. 1994.

(a) (b)

(c) (d)

Figure 4: Hybrid fractal-DCT 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).