GENERALIZED IMAGE CODING USING FRACTAL-BASED METHODS

Mohammad Gharavi-Alkhansari and Thomas S. Huang

Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign
405 N. Mathews Ave., Urbana, Illinois, U.S.A. 61801

ABSTRACT

A new general method is proposed for image coding which exploits similarities, possibly with scaling, among different parts of the image. The coding is performed by approximating each image block with a linear combination of blocks selected from a pool of basis blocks. This pool is made up of (1) a set of fixed basis blocks, (2) a set of blocks taken from the filtered, subsampled image, and (3) a set of blocks taken from the image without any change of scale. When the last two sets are selected causally, the decoding process is noniterative with no constraints on the coefficients of the basis blocks. The index of the selected basis blocks and their corresponding coefficients make the code for each range block. Methods are proposed for making the pool and selecting blocks from the pool.

1. INTRODUCTION

Since the introduction of the concept of fractals by Mandelbrot in late 70's and early 80's [1] it has been exploited in many areas of science and engineering. Barnsley [2] proposed to use fractal properties of natural images for image compression. Barnsley's work was used by Jacquin [3] who developed an algorithm for automatic compression of images. The work of Barnsley and Jacquin has made a basis for further development of fractal-based methods by other researchers, for compression of both still images [4, 5, 6, 7, 8, 9] and image sequences [10, 11].

The essence of most fractal-based coding methods is to approximate each segment of the image by applying a (contractive) transformation on some bigger segment in the image. Then one can reconstruct the image (with some error) by only using the parameters of the transformations. These methods are typically based on the work by Barnsley and Jacquin on *Recursive Iterated Function Systems (RIFS)* [2, 12, 3] and they basically encode most of the information in the image by coding relations among different segments (of different sizes) of the image.

In this paper we introduce a new coding method for images which uses fractal techniques with multiple domain blocks and multiple fixed basis blocks. The result is a generalized image coding method.

2. THE NEW METHOD

2.1. ENCODING PROCESS

In our method, the image is first partitioned into non-overlapping square range blocks of size $S_R \times S_R$. Each range block is considered as an $N = S_R^2$ dimensional vector. For each range block, a pool of $S_R \times S_R$ basis blocks[1] is made and a small number of them are chosen such that their linear combination gives a good approximation of the range block. The index of these chosen basis blocks and their corresponding coefficients constitute the code for the range block. It is notable that the pool of basis blocks can be different from one range block to another, and the range-block-dependent part of the pool is neither necessary nor sent for the decoding process and is not part of the code.

2.1.1. MAKING THE POOL OF BASIS BLOCKS

For each range block, the pool of the basis blocks is made up of the following subsets,

1. *Adaptive Basis Blocks (ABB)*:

   This set contains basis blocks that are generated by applying some transformation T on domain blocks in a neighborhood of the range block. This set is range-block dependent, i.e.,

This research was supported by a grant from Mitsubishi Electric.

[1] In the literature, the expression "basis vectors" usually refers to linearly independent vectors that span a space. However, in this paper, basis blocks (vectors) are not necessarily linearly independent, but we use the expression basis blocks because of the way we use them to approximate range blocks.

it may be different from one range block to another. The set of ABBs itself consists of two subsets,

(a) *Higher Scale Basis Blocks (HSBB)*:
These are blocks that are generated by shrinking domain blocks of size $S_D \times S_D$ of the image ($S_D > S_R$) to give $S_R \times S_R$ basis blocks. The set may also be complemented by adding rotated or reflected versions of the above blocks.

(b) *Same Scale Basis Blocks (SSBB)*:
These are blocks that are directly taken from the image (with no shrinking) and are restricted to be in parts of the image that are already encoded (selected causally). This subset may again be complemented by adding rotated or reflected versions of themselves.

2. *Fixed Basis Blocks (FBB)*: This set contains a series of blocks which are independent of the range block being encoded and is designed by the encoder designer. The set of FBBs is the same for all the range blocks in the image being encoded, and must be sent to the encoder offline. The presence of these basis blocks in the pool can serve the following purposes,

   - The encoder can use shorter codes for the indices of some of these blocks that are commonly present in the range blocks in the form of a strong component.
   - Enable the decoder to encode more accurately range blocks which are very different from other blocks in the image and therefore cannot be well-encoded by other subsets of the pool.
   - Allow the linear combination of basis blocks to be a contractive transformation.

2.1.2. FINDING THE BEST SET OF BASIS BLOCKS

For each range block, after the set of basis blocks is constructed, we look for the smallest number of basis blocks that, when linearly combined, can approximate the range block within a given small error range. We look for the minimum number of basis blocks, because this results in the shortest code for the range block. In this selection, we may put a limit on the maximum number of basis blocks which may be used from either subsets of the basis blocks.

The above problem is an integer programming optimization, and as our basis blocks are not orthogonal, finding the absolute optimum seems to be a rather difficult problem. However, we can use a suboptimal solution to this problem. We propose two different solutions:

1. Choose the basis block which has the strongest correlation coefficient (highest absolute value) with the range block. Then remove any component of its form from the range block and repeat this process for the residual of the range block with the rest of basis blocks until the residual becomes smaller than a threshold or until no other basis block has significant correlation with the residual range block.

2. Same as first method, with the difference that after a basis block is selected, remove any component of its form not only from the range block, but also from all other basis blocks before repeating the process. The residual of range block is computed at each step using standard least squares methods over all the chosen basis blocks.

   It is easy to prove that at each single step, this method can never perform worse than the first method in reducing the norm of the residual of the range block. However, this does not guarantee a better performance in the over all optimization.

We refer to the above methods as *decomposition without basis orthogonalization* and *decomposition with basis orthogonalization* accordingly.

It is notable that most other fractal-based image coding methods are special cases of this method where maximum number of FBBs is set to 1, and no SSBBs are allowed. If the maximum number of ABBs is set to 0 and the maximum number of FBBs is set to $N = S_R^2$, and the FBBs are selected orthogonal, this method reduces to a block transform coding. On the other hand if the maximum number of ABBs is again set to 0 and the maximum number of FBBs is set to 1, then this method reduces to a vector quantization method.

2.2. DECODING PROCESS

The decoding algorithm, in the most general case, is similar to the one proposed by Jacquin [3] and is based on the Contraction Mapping Theorem. We begin with any initial image, and for each range block bring the blocks from the image that have the same address as the selected domain blocks, and apply

Table 1: Paremeter settings and results for the second set of experiments

Total number of basis blocks	Number of FBBs	Number of SSBBs	Number of HSBBs	rms error	Average number of basis blocks used
64	6	25	33	4.91	5.59
64	64	0	0	4.96	5.15
256	6	0	250	4.82	4.25
256	6	125	125	4.78	3.46
256	6	250	0	4.86	3.92
256	64	0	192	4.82	3.78
256	64	96	96	4.79	3.43
256	64	192	0	4.86	3.75

the corresponding transformations on these blocks to make an approximation of the selected ABBs. Then, these approximated basis blocks and the selected FBBs are multipled by their corresponding coefficients and are added together to make an approximation of the range block. This process is repeated for all range blocks until the resulting image does not change significantly with iterations.

The convergence of the decoding process is proven only for the case where the maximum allowed number of ABBs is 1 and their combination coefficients are less than 1. However, experimental results suggest that the decoder converges even when the maximum number of allowed ABBs is greater than 1 [8].

As mentioned before, the SSBBs are restricted to be chosen causally from the image, but the HSBBs are not. An interesting case occurs if we restrict the HSBBs to be also chosen causally. Then the whole encoding system becomes causal and the decoding process needs only a single iteration to converge and there are no restrictions on the coefficients of the basis blocks (except due to possible numerical stability issues).

3. EXPERIMENTAL RESULTS

The encoding algorithm proposed in this paper was applied to Lena image. The settings of the parameters of encoder are as follows,

- $S_R = 8$ and $S_D = 16$.
- Two different sets of FBBs were used in the experiments, only one of which is used in each experiment. The first set is made up of six mutually orthogonal blocks basically of the form $z = 1$, $z = x$, $z = y$, $z = x^2 - a$, $z = y^2 - a$, and $z = xy$, where the z variable represents the pixel value and the origin of the x and y coordinates is the center of the block. a is a constant and its value depends on the size of the block. In the second set, the 64 orthogonal basis blocks of DCT were used. In both cases, no restriction is put on the maximum number of FBBs used.
- SSBBs are taken from a square with the range block located at its center, while avoiding parts of this square that are not encoded yet. The size of this square depends on the number of SSBBs. If parts of this square fall out of the image, those parts are not used.
- HSBBs are again taken from a square (with possibly a different size) with the range block located at its center. The size of the square depends on the number of HSBBs. If parts of this square fall out of the image, those parts are not used.
- No pixel shufflings are applied on the ABBs.
- Step size for bringing basis blocks from the image is 1.
- Minimum rms of the residual of the range block for stopping selection of basis blocks is set to 6.

In one set of experiments, the encoding process was applied to the 256×256 Lena image with both methods of decomposition without basis orthogonalization, and decomposition with basis orthogonalization for choosing basis blocks. In these experiments, the set of 6 FBBs were used, with 81 HSBBs and no SSBBs. For the case of decomposition without basis orthogonalization, an average of 10.14 basis blocks were needed for encoding range blocks, resulting in an encoding rms error of 5.16 (PSNR 34 dB). For the case of decomposition with basis orthogonalization, an average of 8.88 basis blocks were needed for encoding range blocks resulting in an encoding rms error of 5.10 (PSNR 34 dB). These results show that the method of decomposition with basis orthogonalization gives a shorter code for almost the same PSNR for the 256×256 Lena image.

In another set of experiments, the encoding process was applied to the 512×512 Lena image using only decomposition with orthogonalization. The settings for these experiments and their results are shown in Table 1. As it can be seen from the table, in the experiments the differences in rms error resulting from the encoding process is small (due to a fixed rms thresholding of 6). But considering the average number of basis blocks used for encoding each range block, the results suggest that using a combination of HSBBs and SSBBs gives a better performance compared to using HSBBs alone or SSBBs alone. The results also show that for a given total number of basis blocks, using DCT basis blocks for FBBs gives a better performance compared to using 6 FBBs. Also, complementing the 64 DCT FBBs with the HSBBs and SSBBs reduces the average number of basis blocks that are needed for encoding range blocks, when compared with using DCT FBBs alone. But when the total number of basis blocks is fixed at 64, using the 64 orthogonal DCT FBBs gives a better performance compared to combining 6 FBBs with HSBBs and SSBBs.

4. CONCLUSIONS

In this paper we presented a new generalized image compression method. The block transform coding, standard VQ, and most of the earlier fractal compression methods can be considered as special cases of this method. In this method, the fixed basis blocks are complemented with a set of adaptive basis blocks. These additional basis blocks are constructed from the image itself; some from larger scales and some from the same scale as of the range block. The proposed method exploits self-similarities of image both at different scales and at the same scale. Two methods were tested for solving the discrete optimization problem of choosing the smallest number of basis blocks that can closely approximate each range block. It was found that the method of decomposition with basis orthogonalization gives a better performance compared to the method of decomposition without basis orthogonalization. Also, results of using two different sets of fixed basis blocks, and also using a combination of same-scale basis blocks and higher-scale basis blocks were given.

5. REFERENCES

[1] B. B. Mandelbrot, *The Fractal Geometry of Nature.* New York: W. H. Freeman and Company, 1982.

[2] M. F. Barnsley and H. Rising III, *Fractals Everywhere.* Boston: Academic Press Professional, second ed., 1993.

[3] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Transactions on Image Processing*, vol. 1, pp. 18–30, Jan. 1992.

[4] Y. Fisher, ed., *Fractal Image Compression: Theory and Application.* New York: Springer-Verlag, 1995.

[5] Y. Fisher, "Fractal image compression," in *SIGGRAPH '92 Course Notes: From Folk Art to Hyperreality* (P. Prusinkiewicz, ed.), pp. 1–21, 1992.

[6] G. E. Øien, *L_2-Optimal Attractor Image Coding With Fast Decoder Convergence.* PhD thesis, The Norwegian Institute of Technology, Norway, Apr. 1993.

[7] D. M. Monro, "A hybrid fractal transform," in *Proceedings of IEEE ICASSP-93*, vol. V, (Minneapolis, Minnesota), pp. 169–172, Apr. 27–30, 1993.

[8] M. Gharavi-Alkhansari and T. S. Huang, "A fractal-based image block-coding algorithm," in *Proceedings of IEEE ICASSP-93*, vol. V, (Minneapolis, Minnesota), pp. 345–348, Apr. 27–30, 1993.

[9] G. Vines and M. H. Hayes, III, "Adaptive IFS image coding with proximity maps," in *Proceedings of IEEE ICASSP-93*, vol. V, (Minneapolis, Minnesota), pp. 349–352, Apr. 27–30, 1993.

[10] L. P. Hurd, M. A. Gustavus, and M. F. Barnsley, "Fractal video compression," in *Digest of Papers. Thirty-Seventh IEEE Computer Society International Conference (COMPCON)*, pp. 41–42, Feb. 24–28, 1992.

[11] H. Li, M. Novak, and R. Forchheimer, "Fractal-based image sequence compression scheme," *Optical Engineering*, vol. 32, pp. 1588–1595, July 1993.

[12] M. F. Barnsley and L. P. Hurd, *Fractal Image Compression.* Wellesley, Massachusetts: AK Peters, Ltd., 1993.