IFS CODING USING AN MPC NETWORK LIBRARY

Robert D. Dony[1] *Edward R. Vrscay*[2]

[1]School of Engineering
University of Guelph
Guelph, ON N1G 2W1
email: rdony@uoguelph.ca

[2]Deptartment of Applied Mathematics
University of Waterloo
Waterloo, ON N2L 2G1
email: ervrscay@links.uwaterloo.ca

ABSTRACT

This paper examines the relationship between iterated function systems (IFS), a fractal approach to image compression, and mixutre of principal components (MPC), a neural network approach to image compression. Both can be fundamentally expressed as a local linear transformation. In IFS, the basis vector comes from the image itself and evolves during the iterations while in MPC, the trained network contains the basis basis vectors. A new method of image compression is presented which uses an MPC network as a library for reducing the search for the large domain blocks in IFS. The resulting hybrid approach has better rate-distortion characteristics relative to standard IFS when tested on a standard image.

1. INTRODUCTION

A number of new nonlinear techniques for image compression have emerged recently which include fractal methods [1, 2, 3] and neural network methods [4]. They are nonlinear approaches and have been shown to have advantages over standard techniques. However, for fractal coding, the computational requirements for encoding an image are significant. This paper presents a new method which combines features of both approaches, resulting in improved performance.

2. MIXTURE OF PRINCIPAL COMPONENTS

A new approach to data representation, a mixture of principal components (MPC), has recently been developed [5, 6]. The MPC representation has been used to develop a number of novel neural network-based adaptive image compression methods. It is a modular architecture where each module consists of a set of one or more basis vectors which performs a linear transformation on the input data. In addition, each module represents a class of input data and the basis vectors of each module defines the class.

Figure 1 illustrates the architecture of the network. The input vector, $\mathbf{x} \in \mathbf{R}^N$, is linearly transformed by each of the K modules resulting in K coefficient vectors, $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_K\}$. Each module contains an orthonormal set of M basis vectors defining an M-dimensional linear subspace. So, if $\mathbf{W}_i \in \mathbf{R}^{N\times M}$ is an $N \times M$ matrix whose M columns contain the M basis vectors, the coefficient vector $\mathbf{y}_i \in \mathbf{R}^M$ is calculated as

$$\mathbf{y}_i = \mathbf{W}_i^T \mathbf{x} \qquad (1)$$

The classifier then chooses the output of the winning module based on the subspace classifier

$$\arg\left\{\max_i(\|\mathbf{P}_i\mathbf{x}\|)\right\} \qquad (2)$$

where $\mathbf{P}_i \in \mathbf{R}^{N\times N}$ is the linear projection matrix calculated as

$$\mathbf{P}_i = \mathbf{W}_i\mathbf{W}_i^T \qquad (3)$$

Since $\mathbf{W}_i$ is an orthonormal set, *i.e.*, $\mathbf{W}_i^T\mathbf{W}_i = \mathbf{I}$, an equivalent classifier to 2 is

$$\arg\left\{\max_i(\|\mathbf{y}_i\|)\right\} \qquad (4)$$

The winning class has the largest coefficient vector norm. The network then outputs the winning class index, k, and the corresponding coefficient vector, $\mathbf{y}_k$.

The decoder has the same set of transformations, $\{\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_K\}$. The reconstructed vector, $\hat{\mathbf{x}}$, is calculated as

$$\hat{\mathbf{x}} = \mathbf{W}_k\mathbf{y}_k \qquad (5)$$

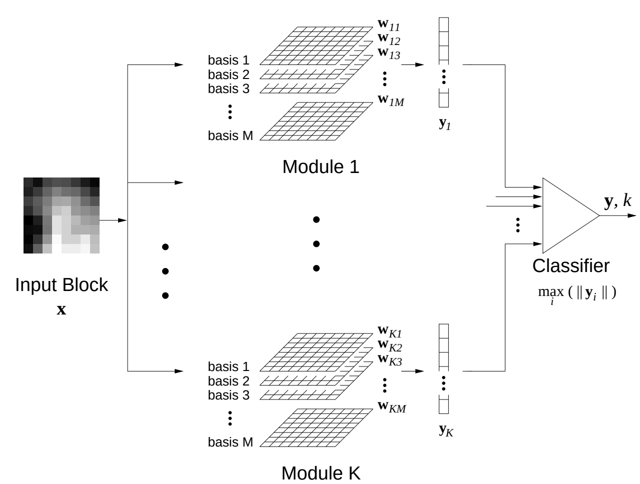

Figure 1. Coding section for MPC network with K classes and M components per subspace.

The performance of these adaptive networks can surpass that of the optimal nonadaptive Karhunen Loève transform (KLT); up to a 3 dB gain in signal-to-noise can be realized [5]. Also, the networks have significant computational advantages in terms of complexity at the decoder over the KLT

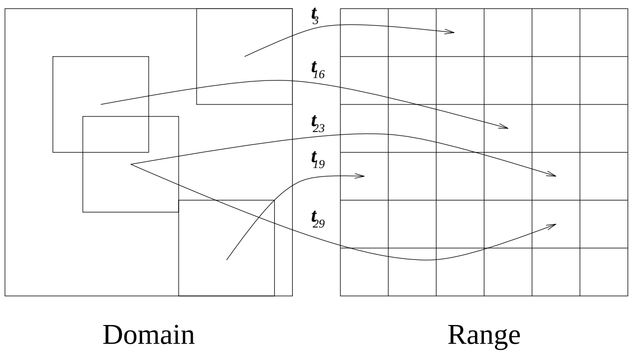

Figure 2. In an IFS, self-similarity is sought between the larger domain blocks and the smaller range blocks in an image. Associated with each range block is a transformation, $\mathbf{t}_i$, which specifies the self-similar domain block. (From [11])

and the fast discrete cosine transform (DCT). When applied to the problem of compressing digital chest radiographs, it was found that compression ratios of between 30:1 and 40:1 can be realized without an unacceptable loss in image quality [7]. Furthermore, the distortion introduced by the new method was judged to be generally less objectionable than that from the KLT. Similar advantages have been demonstrated for synthetic aperture radar (SAR) images [8].

3. ITERATED FUNCTION SYSTEMS

An emerging technique in image compression is the use of Iterated Function Systems (IFS) to compactly represent graphics or images [1, 2, 3, 9]-[15]. With IFS, an image is represented by a set of mappings which, in general, through a contractive affine transformation maps larger regions of an image to self-similar smaller regions. For practical image compression, a number of restrictions on the mapping functions are imposed. Typically, the regions are image blocks of a fixed size, the ratio between the larger and smaller blocks is fixed at two, the blocks may only undergo translation followed by one of eight symmetry operations (*e.g.*, rotation by $n \times 90°$, or a reflection), and the grey level value gets shifted and scaled.

This process is illustrated in figure 2. Each range block $\hat{\mathbf{x}}_i$ has associated with it a transformation $\mathbf{t}_i$ which specifies: a translation, $(x, y)_i$, a symmetry operator, a grey level scale factor, a_i, and a grey level offset, b_i. It is the set of transformations, $T = \{\mathbf{t}_i | i = 1, \ldots, n\}$, which are stored or transmitted as the compressed representation. For reconstruction, an iterative process is performed. Initially, the domain image is set to some arbitrary value (*e.g.*, a solid grey level). The range image is constructed using the set of transformations on the domain image. For the next iteration, the new range image then becomes the domain image and the process is repeated. The range image converges to what is referred to as the "attractor image" as the number of iterations increases. For a good set of transformations, the attractor image is sufficiently close to the original image.

The operations to calculate a range block, $\hat{\mathbf{x}}_i^{(n+1)}$, at an iteration $n+1$ can be expressed as follows. Let $\hat{f}^{(n)}(x, y)$ be the image generated after n previous iterations. Let $H_{(x,y)_i}$ be the sampling operator for the block that extracts the larger domain block at location (x, y) in an image. Let Θ_i be the symmetry operator for the block that may rotate or mirror the domain block. Let $\downarrow_2$ be the operator that reduces the size of the block by a factor of two. Then, let $\mathbf{z}_i^{(n)}$ be the range block before the scaling and offset is applied, formed as

$$\mathbf{z}_i^{(n)} = \downarrow_2 \cdot \Theta_i \cdot H_{(x,y)_i} \cdot \hat{f}^{(n)}(x, y) \tag{6}$$

Let a_i be the scaling factor for the block and let $\mathbf{b}_i$ be a vector containing N offset factors, b_i, for the block where N is the number of elements in the block, *i.e.*, $\mathbf{b}_i = [b_i \; b_i \; \ldots \; b_i]^T$. The range block is then calculated as

$$\hat{\mathbf{x}}_i^{(n+1)} = a_i \mathbf{z}_i^{(n)} + \mathbf{b}_i \tag{7}$$

The interesting problem is that of determining the appropriate transformation set for a given image. An exhaustive search of the entire transformation space for each range block to determine the optimal transformation is impractical due to the large computational requirements. Therefore, an active area of research is in determining computationally efficient methods of searching the transformation space to calculate the set T for an image [12, 2, 13].

4. RELATIONSHIP BETWEEN MPC AND IFS

To illustrate the relationship between MPC and IFS, let us rewrite equation 7 as

$$\hat{\mathbf{x}}_i^{(n+1)} = \alpha_i \mathbf{w}_{i,2}^{(n)} + \beta_i \mathbf{w}_1 \tag{8}$$

or equivalently

$$\hat{\mathbf{x}}_i^{(n+1)} = \mathbf{W}_i^{(n)} \mathbf{y}_i \tag{9}$$

where

$$\mathbf{y}_i = \begin{bmatrix} \beta_i \\ \alpha_i \end{bmatrix} \tag{10}$$

and

$$\mathbf{W}_i^{(n)} = \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_{i,2}^{(n)} \end{bmatrix} \tag{11}$$

In other words, the IFS equation 7 is of the same form as that of the MPC equation 5.

It follows, then, that $\mathbf{w}_1$ is simply DC, *i.e.*,

$$\mathbf{w}_1 = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \tag{12}$$

For $\mathbf{W}$ to be an orthonormal system, then the basis vector $\mathbf{w}_{i,2}^{(n)}$ is derived as

$$\mathbf{w}_{i,2}^{(n)} = \frac{\mathbf{z}_i^{(n)} - (\mathbf{w}_1^T \mathbf{z}_i^{(n)})\mathbf{w}_1}{\|\mathbf{z}_i^{(n)} - (\mathbf{w}_1^T \mathbf{z}_i^{(n)})\mathbf{w}_1\|} \tag{13}$$

Both representations are based on linear transform coding. The essential difference between the two methods, is that for MPC the basis vectors are computed using principal components analysis from training sessions on sample images while the single basis vector (recall that $\mathbf{w}_1$ is fixed at DC while $\mathbf{w}_{i,2}^{(n)}$ is variable) for IFS is generated from a portion of the image being reconstructed.

5. IFS CODING USING AN MPC NETWORK LIBRARY

Disregarding the symmetry operator, Θ_i, for the sake of simplicity, each image block under IFS is represented as the (x, y) coordinate of the larger domain block used to form the transformation basis set $\mathbf{W}$ at the given iteration and the two-dimensional coefficient vector $\mathbf{y}$. However, given an exhaustive search, numerous domain blocks may form adequate basis vectors for a given range block. It may be advantageous then to collect a representative set of domain block locations prior to coding and limit the search to only that set to determine the IFS mapping. The training algorithms for MPC networks can be used to generate such a set.

This paper presents a hybrid technique which combines both IFS and MPC. Initially, an MPC network of subspace dimension $M = 2$ with the first component fixed at DC and the second allowed to vary and K classes is trained on a "typical" image or set of images. This network is then used to form a library of a limited number of locations (K of them) for choosing the large sized blocks for the mapping. Initially, the image to be coded is searched to find the location of the best match in the image for each MPC module. Then, during coding, only those K locations in the image are searched to determine the large block used for the affine transform. This approach saves both search time and bits required for encoding the location.

6. METHOD

6.1. MPC Network Training

Using training methods similar to [8], a number of MPC networks were trained on a set of standard images. The block size chosen was 8×8. This set did *not* include the standard "Lenna" image which was used later for evaluation. The number of classes was grown from 1 to 1024 by doubling the number of classes. In each case, the number of components was fixed at one common DC component plus a single, second component per class.

6.2. IFS Coding

In IFS coding, the mapping from the larger domain block to the smaller range block may be specified by the (x, y) coordinate, a symmetry code, a grey level offset and a grey level scale factor. For this evaluation, no symmetry transformation was performed and the scale and offset values were not quantized. To vary the representation, the spacing of the grid from which the larger domain block were chosen was varied, including grid step spacings of 2, 4, 8, 16, and 32. The domain block size was 16×16 and the range block size was 8×8. For each of these 5 representations for the test image, the MSE was recorded after the system converged.

6.3. IFS Coding Using an MPC Network Library

Each of the trained MPC networks were used as the library for the hybrid IFS/MPC method. For a given network, each class basis set was matched against each possible domain block (a grid spacing of 1, *i.e.,* an exhaustive search) reduced to the range block size (8×8). The (x, y) coordinate of the domain block which best matched the class was recorded for the class. A subspace classifier was used to choose the winning block. The resulting set of class, coordinate pairs were then used as the library of domain blocks to be searched to construct the self-affine mapping for the IFS representation. This library is stored with the subsequently encoded representation of the image. In constructing the IFS, only those domain blocks in the library are searched for each range block. Instead of recording the coordinates of each domain block, only its index in the library is stored. As above, no quantization was performed on the scale and offset coefficients.

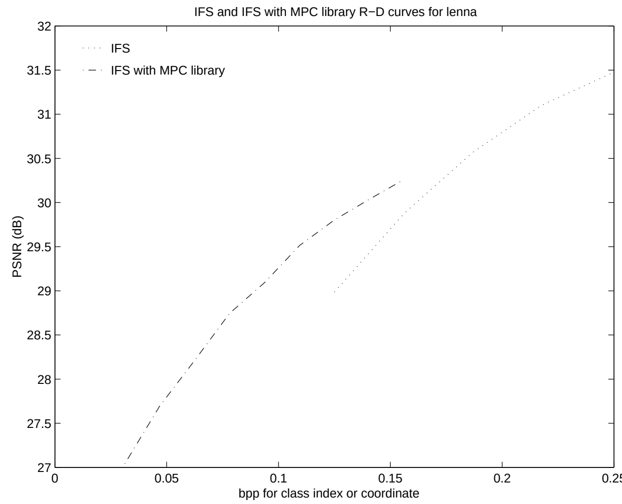

Figure 3. Rate-distortion curves for each of the three techniques. The rate is the number of bits per pixel required to encode the class index for IFS using an MPC library, or the coordinates on the domain grid of the domain block for standard IFS.

7. RESULTS

Figure 3 shows the rate-distortion curves for the two techniques. Since the two coefficients corresponding to the scale and offset for the two IFS methods were not quantized, only the coordinate information determined the base bit rate.

When comparing the two IFS approaches, the figure shows that for the same number of bits required for encoding either the coordinates of the domain blocks or the index of the MPC library for each range block, the use of the hybrid IFS method results in less distortion. For example, at an index or coordinate bit rate of 0.125 bits per pixel, the standard IFS had a PSNR of 29.0 dB while the hybrid IFS method's PSNR was 29.8 dB.

Figures 4 and 5 show the results at 0.125 bpp for the standard IFS and hybrid IFS with MPC library coding methods respectively. The hybrid IFS method shows less distortion than the standard IFS. For example, the block effect distortion is less noticeable in smooth regions with the IFS/MPC method. Also, the edge of the shoulder in the bottom right of the image is preserved under the IFS/MPC method while the IFS approach introduces significant distortion. This difference in quality agree with the numerical error results shown above.

8. CONCLUSION

This paper illustrates the conceptual similarities between the IFS representation and the MPC representation for image coding. IFS is in effect a transform coding method

Figure 4. IFS coding at 0.125 bpp for coordinate encoding.

Figure 5. IFS with MPC library coding at 0.125 bpp for index encoding.

where the basis vectors are generated from a portion of the image being reconstructed. By exploiting this similarity, a new method had been developed which uses a library of domain blocks generated by MPC in an IFS scheme. This new method appears to result in significant improvements over the standard IFS method. Both the search complexity is reduced as well as the distortion.

REFERENCES

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

[2] A. E. Jacquin, "Fractal image coding: A review," *Proc. IEEE*, vol. 81, pp. 1451–1465, Oct. 1993.

[3] B. Forte and E. R. Vrscay, "Theory of generalized fractal transforms," in *Fractal Encoding and Analysis* (Y. Fisher, ed.), Springer Verlag, 1998.

[4] R. D. Dony and S. Haykin, "Neural network approaches to image compression," *Proc. IEEE*, vol. 83, pp. 288–303, February 1995.

[5] R. D. Dony and S. Haykin, "Optimally adaptive transform coding," *IEEE Trans. Image Processing*, vol. 4, pp. 1358–1370, October 1995.

[6] R. D. Dony and S. Haykin, "Multi-class maximum entropy coder," in *1995 IEEE Int. Conf. Systems, Man, and Cybernetics*, (Vancouver, B.C.), October 22-25 1995.

[7] R. D. Dony, S. Haykin, C. Coblentz, and C. Nahmias, "Compression of digital chest radiographs using a mixture of principal components neural network: An evaluation of performance," *RadioGraphics*, vol. 16, November 1996.

[8] R. D. Dony and S. Haykin, "Compression of SAR images using KLT, VQ, and mixture of principal components," *IEE Proceedings – Radar, Sonar and Navigation*, vol. 144, pp. 113–120, June 1997.

[9] E. R. Vrscay, "Iterated function systems: Theory, applications and the inverse problem," in *NATO ASI Series C* (E. Kaltofen and S. Watt, eds.), vol. 346, pp. 404–468, 1991.

[10] 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.

[11] J. Domaszewicz and V. A. Vaishampayan, "Structural limitations of self-affine and partially self-affine fractal compression," in *Visual Communications and Image Processing '93* (B. G. Haskell and H.-M. Hang, eds.), vol. 2094 of *SPIE Proceedings*, pp. 1498–1507, 1993.

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

[13] J. Kominek, "Advances in fractal compression for multimedia applications." to appear in *Multimedia Systems Journal.*

[14] N. Lu, *Fractal Imaging.* NY: Academic Press, 1997.

[15] E. R. Vrscay, "A generalized class of fractal-wavelet transforms for image representation and compression," To appear in *Can. J. Elect. Comp. Eng.*, vol. 23, no. 1,2, 1998.