Multiscale fractal image coding and the two-scale difference equation

Alexandru Bogdan
Department of Electrical Engineering and Center for Telecommunications Research
Columbia University, New York, N.Y. 10027

Abstract

Decoding a fractal compressed image can be seen as solving an affine two-scale functional equation. If the affine term is zero, the dilation equation becomes linear and the solution is similar to the scaling functions from wavelet theory. In both cases a linear operator, parameterized with a few variables determines a complicate looking function. We present two classes of algorithms for gray image coding, based on the solutions of the affine and of the linear equation. The affine algorithm is used to build a multiscale pyramid coding scheme. It has been applied to the coding of gray images and also to color image compression and interframe video coding.

I. INTRODUCTION

The fractal image coding algorithm introduced by A. Jacquin [1, 2] can be described using fixed-point theory. The original image f is approximated by the fixed-point of a transformation T from $\mathcal{X}$ the space of gray images with the same support, onto itself: $f = Tf$. The mapping T is the fractal code and it is restricted to the class of affine mappings: $Tf = Af + b; f, b \in \mathcal{X}$ and A is a linear operator. The mapping T has an unique fixed-point f_T that can be obtained by solving iteratively the functional equation $f = Af + b$, when T is contractive or equivalently $||A|| < 1$ [3]. To take advantage of the similarity between two different scale representations of the image f the linear operator A acts on the dilated version of the signal (dilation is a linear operation):

$$f(x) = Tf(x) = A_L f(2x) + b \tag{1}$$

where x is the space coordinate for the images. A_L and b can be parameterized with a small number of variables $A_L(\alpha)$ and $b(\beta)$. Coding becomes an optimization problem which minimizes the distortion $||f - f_T||$ or the size of the code $description(T)$ over the parameter space, depending of the desired goal. Good results in image coding have been obtained using partial self-similarity [1, 4, 5, 6] to create a collection of piecewise transformations $\{T_i\}$ which together determine T: $T = \sum_i T_i$. Each T_i determines the individual fractal code for a block of pixels from the image f by mapping a piece from $f(2x)$ onto that block. Note that having the map T_j for a single block is not sufficient to decode the block in this case and we need the whole T. Coding is a computationally intensive operation which requires exhaustive search in the parameter space for the best map $T(\alpha, \beta)$. At this time there is no direct algorithm to generate the transformation T.

The functional equation (1) in this general form is a two-scale affine difference equation. In numerical applications the piecewise self-similarity makes the linear operator A_L very sparse and the iterative procedure of solving (1) is simpler than a direct method which requires the inversion of a large matrix. If $||A_L|| \ll 1$ the solution requires only a few iterations. If the term b in (1) is zero (the image with all pixels a 0), the contractivity of A_L will give the trivial 0 image in $\mathcal{X}$ as a solution when using the iterative method. Iteration will work only for the case when $||A_L||$ admits 1 as a simple eigenvalue and the solution obtained is up to a multiplicative constant, the associated eigenvector. The case when A_L is a convolution operator appears as the lattice two-scale difference equation in wavelet theory [7] and subdivision schemes in computer graphics [8].

$$f(x) = \sum_k c_k f(2x - \beta_k) \tag{2}$$

The coefficients c_k in equation (2) represent a discrete linear operator and may determine an unique solution which is known as the scaling (mother) function. The scaling functions are used in wavelet theory, through translation and dilation, to build a basis in function spaces and a multiresolution analysis [9, 10]. The coefficients $\{c_k\}$ can be seen as the code for the signal $f(x)$. Solving (2) is equivalent to the eigenvalue problem in operator theory [11] which has an unique solution when the linear operator C determined by $\{c_k\}$ has a simple dominant eigenvalue.

Section II. presents the two-scale functional equations used in data compression algorithms. A signal can be decomposed into an affine term and a self-similar part. The contribution of the self-similar part in $f(x)$ depends on the norm of the operator $||A_L||$ in (1). We show that the reconstruction does not have to be iterative but it requires the inversion of a huge matrix in numerical applications. Section III. describes the application of the affine and linear iterative coding algorithms to image compression. We present iterative fixed-point reconstruction algorithms for both the affine and the linear case. We use the affine equation to generate a multiscale pyramid coding algorithm. At each level the affine term is the large scale image and the fixed-point solution of (2) represents the image with the added detail [12] which is also the affine term in the next level. We show how to extend this algorithm to color images and video coding.

This compression method works based on the self-similarity at different scales found in the signal to be coded. Wavelets use full self-similarity since the mother wavelet is a linear combination of scaled copies of itself. The fractal image coding algorithm uses piecewise self-similarity where a piece of $f(x)$ is similar to a region of $f(2x)$. This allows the encoding of much more complicated signals than the full self-similarity approach.

Fractal image coding has been pioneered by Barnsley [13] and Jacquin has introduced the piecewise affine algorithm [1]. The affine term has been originally used as a shift in the gray levels and only recently the independent coding of these blocks using VQ and DCT was mentioned [14]. The tiling of the image plane can be arbitrary, with blocks of pixels of different sizes and shapes [2, 4]. The main contribution of this work is the multiscale pyramid image coding algorithm and the connection to the two-scale difference equation and wavelet theory. Also our approach to video coding where we use the previous frame as the starting point for the actual frame is original. Other approaches to fractal video compression [5] use tri-dimensional blocks as an extension to the two-dimensional method or use the previous frame in a VQ fashion.

II. THE DILATION EQUATION

By Banach's fixed-point theorem [3], a contractive operator on a metric space $T : \mathcal{X} \rightarrow \mathcal{X}$ has an unique fixed point that is the solution of the (nonlinear) equation $\{f = Tf, \quad f \in \mathcal{X}\}$ and can be obtained by the successive approximations method: $\{f_{n+1} = Tf_n, \ n = 0, 1, \ldots\}$. When T is affine as in (1), its contractivity is that of the linear operator A_L. If the affine term b is the 0 vector in $\mathcal{X}$ the contractivity of A_L guarantees the unique trivial solution $f(x) = 0$. If A_L is no longer contractive and $\sigma(A_L) = 1$ is a dominant simple eigenvalue, then up to a multiplicative constant, we obtain by iteration the eigenvector associated with $\sigma(A_L)$. In linear algebra this is studied as the matrix eigenvalue problem [11].

Self-similarity is a wide-spread property in nature. This explains the success of wavelets in signal processing, fractal modeling of natural processes and fractal image compression. The two-scale difference equation has been studied extensively and [7] contains a detailed survey of different contexts where it arises. The dilation term $f(2x - \cdot)$ is crucial in the determination of the properties and uniqueness sought for the solutions of (2). As an example in [7] the $L^1(\Re)$ solutions with compact support are studied. Full self-similarity is to restrictive when we want to model arbitrary functions as the solution of a dilation equation. Piecewise self-similarity has been used in the fractal image compression algorithm and the associated dilation equation becomes:

$$f(x) = \sum_k c_k w(x-k) f(2x - \beta_k) \tag{3}$$

where $w(x-k)$ is a window function, as an example the two-dimensional Haar function in Jacquin's algorithm. Equation (3) can be also written in the affine form (1). Our goal is to decompose a signal into a self-similar part controlled by the linear operator A_L as in (1) and the affine term b. We say that the signal is completely self-similar if the affine term b is 0. To see how this decomposition is influenced by the relative size of the two parts we study the solutions of the followig affine dilation equation:

$$f(x) = \gamma A_L f(2x) + b \tag{4}$$

where A_L determines the Daubechies-4 mother wavelet and b is a Hanning windowing function. The factor $\gamma < 1$ weigths the contribution of the self-similar part in the solution.

In Fig.1 we present the Daubechies-4 mother wavelet which is the solution of (2) having four coefficients different from zero, and the Hanning windowing function we chose for the affine term. Both have been scaled to have equal norm. On the right we present the iterative solution of the affine equation with $\gamma = .98$. The result is very close to the self-similar wavelet except for a multiplicative factor. This is to be expected because for $\gamma = 1$ the solution blows out-of-bound. Fig.2 presents the the solution of (3) for ($\gamma = .1$ to $.5$) on the left and ($\gamma = .6$ and $.7$) on the right, together with the affine term ($\gamma = 0$). In these simulations we see how the degree of self-similarity in the solutions are controlled by the contractivity of the linear operator A_L. Given the affine term b the resulting function is coded with 5 parameters: the factor γ and the 4 coefficients of the

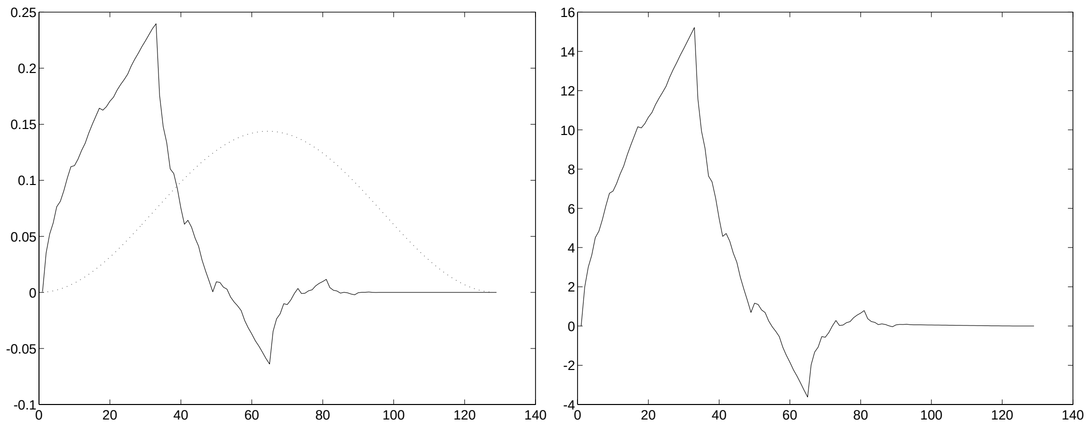

Figure 1: Left: the Daubechies-4 mother wavelet and the Hanning windowing function(...) used as the affine term in the dilation equation, normalized to the same value. On the right we have the iterative solution of the affine two-scale functional equation for $\gamma = .98$.

Daubechies mother wavelet. In fractal coding we have to solve the inverse problem of finding the mapping that has the given signal as a fixed-point.

A. *Pyramid fractal decomposition*

In some applications, like image coding, we are interested in a hierarchy of signals at different scales, which when combined reconstruct the original [12]. Assuming we know how to solve the fractal coding problem, we can build a pyramid hierarchy based on the solutions of equations of type (1). The original signal $f_0(x)$ is decomposed into a self-similar part and an affine term $f_1(x)$. Next f_1 is decomposed into a self-similar part and an affine term $f_2(x)$ and so on.

$$\begin{aligned} f_0(x) &= A_L^0 f_0(2x) + f_1(x) \\ f_1(x) &= A_L^1 f_1(2x) + f_2(x) \\ &\dots \\ f_{n-1}(x) &= A_L^{n-1} f_{n-1}(2x) + f_n(x) \end{aligned} \tag{5}$$

If as n grows, $f_n(x)$ is a coarser and coarser signal representation, we have achieved our goal. One more note on the iterative procedure for signal reconstruction. Since the operator A_L is contractive, it has no eigenvalues equal to one. We can try to solve directly for f(x) in the affine functional

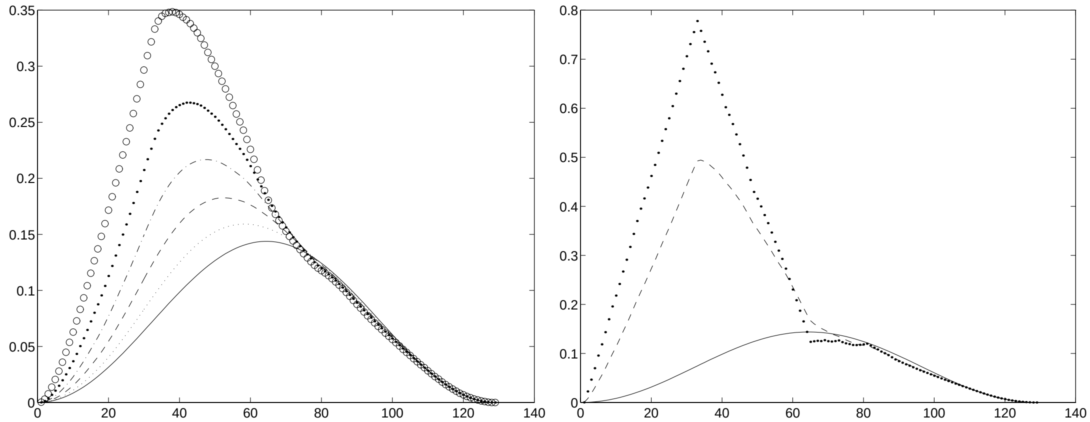

Figure 2: Left: The iterative solution of the affine two-scale functional equation for $\gamma = 0$ (–), $\gamma = .1$ (...),.2 (- -), .3(-.), .4($\cdot$) and .5(o). On th right the solutions for $\gamma = 0$ (–), $\gamma = .6$ (- -) and .7 (...).

equation $f = Af + b$. We obtain $f = (I - A)^{-1}b$ and when we iterate the pyramid decomposition, (5) becomes:

$$f_0 = (I - A^0)^{-1}(I - A^1)^{-1} \dots (I - A^{n-1})^{-1} f_n \tag{6}$$

In numerical applications, this is the problem of solving a system of linear equations of the form $Cy = g$ [15, 16]. In our case the matrix realisation of A is very sparse and fast algorithms can be used in iteration, while $(I - A)$ becomes a full matrix when inverted. To our knowledge, signal analysis of the type (5) have not be studied. In section III. we present such a decomposition used into a pyramid image fractal coding algorithm.

III. APPLICATION TO IMAGE CODING

Our goal is to obtain a multiscale image decomposition of the type (5) where we start with a coarse representation at level n and we add some detail to reach level $n - 1$ [12]. In this work we model the image $f(x)$ as a real function defined on a rectangle in $\Re^2$ so x is a two dimensional coordinate vector. Fractal coding is the process of finding the pair (A_l, b) in equation (1) having a very compact description and which will give a very good approximation for f. Since full self-similarity is very restrictive we are looking for partial self-similarity. Given a tiling of the image support $\cup_i support(r_i) = support(\mathcal{X}) \in \Re^2$, we define the fractal code piecewise: $T = \sum_i T_i$ where

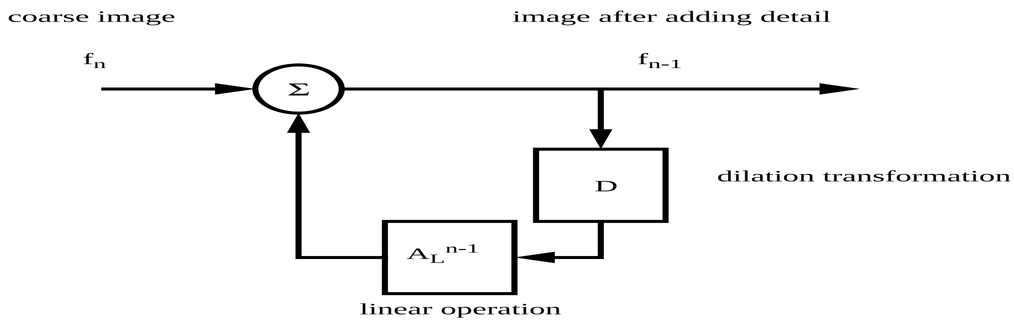

Figure 3: The decoding of level $n-1$ based on level n in a pyramid fractal coding scheme for gray images.

the range of each mapping T_i is supported on $support(r_i)$ and the domain of T_i can be anywhere in $\mathcal{X}$. Each T_i is found individually, usually by exhaustive search in the parameter space, to give the best approximation of the piece $f_i = f|_{r_i} \approx T_i f$ [14]. The parameter space for each T_i contains a gray level multiplicative factor μ_i and the position of the domain block in $f(2x)$ such that a fixed number of bits will code the piecewise transformation independent of the size of the range tile r_i. When we cover $\mathcal{X}$ with a small number of large tiles we obtain high compression but poor approximation because of the limited choice of self-similar patterns. In practice $\mathcal{X}$ is a rectangle tiled with square or rectangular blocks to simplify the description.

A. *Fractal coding of gray images*

We limit our multiscale analysis to 5 levels in our simulation. Level zero represents the original digitized image and level 1 is the self-similar coding with tiles of 2 by 2 pixels in size. The coding starts at level 5 with 32 by 32 pixels tiles. Because f_5 is the top of the pyramid, the affine term b is zero so our code has the form (3). To obtain an iterative solution we have to make sure the mapping has a simple eigenvalue equal to one. This is difficult in applications so we force the decoded image to have the same mean as the original image. We rescale at each iteration using a multiplicative factor, or an additive term that is equivalent to the projection onto the convex space of equal mean images. Next level is the approximation f_4 which is using tiles of 16 by 16

level/ block size	5. 32x32 pels	4. 16x16 pels	3. 8x8 pels	2. 4x4 pels	1. 2x2 pels
coder type	PSNR/bpp	PSNR/bpp	PSNR/bpp	PSNR/bpp	PSNR/bpp
self-similar	22.2dB	26.03dB	29.78dB	34.55dB	—
	.0273bpp	.109bpp	.4375bpp	1.75bpp	
pyramid	22.2dB	25.83dB	29.18dB	32.39dB	32.99dB
multiscale	.0273bpp	.119bpp	.37bpp	1.1062bpp	1.562bpp
JPEG	22.29dB	25.728dB	29.8dB	34.47dB	
	.067bpp	.107bpp	.234bpp	.71bpp	

Table 1: Fractal image coding results: fixed-rate, same size blocks, top; a pyramid fractal coder, middle; JPEG standard at similar bit rates, bottom.

pixels in the self-similar code and f_5 as the affine term in (1). The procedure is repeated up to level 1 which gives a very good approximation of the original. Figure (3) illustrates the decoding at level $n-1$ from level $n-1$ based on the algorithm (5) or on the direct form(6). Level 0 is not coded because now we have a very ill posed problem. The tiles are 1 pixel in size and you can select any domain block. Moreover we no longer have compression. At each level, we select for coding only those range blocks that have a distortion to the original above a preset threshold, so code size is image dependent. We present in Table1 coding results for the 512 by 512 pixels, 256 gray levels image Lenna. The table presents the (PSNR) distortion and bit rates at level 5 to level 1 for the completely self-similar coder (3) (tiling with equal size blocks) and the pyramid coder (5). For comparison, bit rates for similar distortion rates using the $JPEG$ coder are indicated. We see a slight degradation in performance for the multiscale coder that is expected in the pyramid coding schemes. What numbers cannot show is that at low bit rates the fractal image has a natural appearance while the JPEG coder introduces unpleasant blocking artifacts. Also our coders were not optimized and we can increase compression by adding an extra entropy coding stage like the Huffman coder in JPEG.

B. *Coding color images*

We have used the basic coding algorithm (1) in the coding of RGB color images. The green component is individually coded and then it is used as the affine term in coding the red and blue components. Various other combinations with different image formats are currently under investigation. An example of the information flow in the coding of an RGB color image is given in Fig.4. We generate a multiscale representation based on the GREEN component but we could use

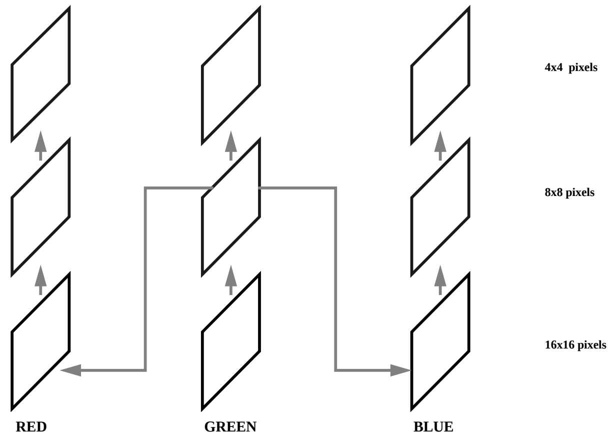

Figure 4: Multiscale fractal coding applied to a RGB color image: information flow.

the RED or the BLUE. The coding gain may depend on our choice of the base of the pyramid but it may change with the image and we have to study this aspects in more detail. In this case the we use 8×8 pixels/block scale level as the affine inputs for the RED and BLUE channels. Empirically we found that the use of a good quality GREEN (small scale) subimage as the base affine term in the other channels will generate better coding gains. This choice can be seen as a design trade-off depending on the specific application .

C. *Low bit-rate video coding*

The superiority of fractal algorithms at high compression rates has been tested in multimedia applications like videoconferencing but with different strategies [17, 5] based mainly on intraframe coding. We are using an approach derived from the pyramid scheme (5) to the coding of video sequences. The previous decoded frame $(n-1)$ is the affine term in (1) and the present frame (n) is the output of a single level decoder as in Fig.3. Different configurations which combine the multiscale and the predictive properties of the algorithm are possible. We present a pyramidal coder with three bit-streams and interframe coding that is a good match for an ATM environment in Fig.5. The base level at low resolution is using (1) for interframe coding. In a telecommunications applications this level uses a very small bandwidth and requires a high level of protection. Additional bit streams can be added at higher resolution if needed by the user or application and

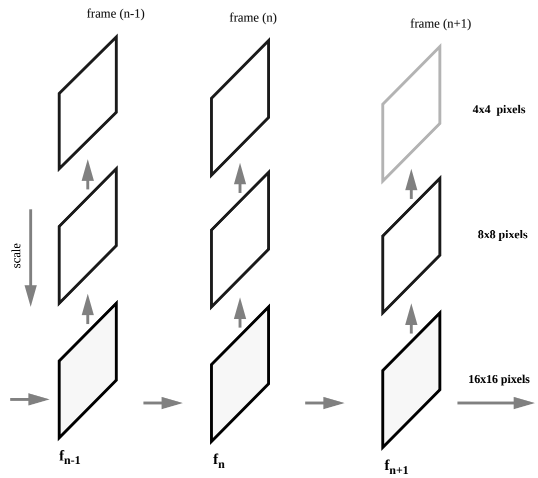

Figure 5: Pyramid fractal video coding with three levels. The base level is the low resolution, low bit rate based on 16x16 pixels blocks. Additional detail is presented in the higher resolution bit-streams and can be added depending on network capacity or user requirements.

if the bandwidth is available.

Simulation has been performed on the well known sequence "Miss America". Quantitative coding results are presented in Fig.6 and Fig.7 and are at least comparable with what has been published. To have a complete fractal coder we have encoded the first frame as a gray image using the multiscale algorithm used in the previous section. We note the advantage of intraframe coding the other frames in the sequence manifested by low bit rates and better fidelity. In fact, to have a good prediction for $frame$ 2 we are using a higher resolution, 8x8 pixels/block fractal code at $frame$ 1 in the base bit stream. These results have been obtained using the pyramid coder for intraframe coding of the base level. We expect to obtain better results when the design of a specialized intraframe coder is finished.

IV. CONCLUSIONS AND FURTHER WORK

We have introduced a multiscale fractal compression algorithm with application to the coding of gray and RGB color images and video sequences. At each level of the coding pyramid, the decoded image is the solution of a two-scale functional equation. This type of equations appear in wavelet theory and subdivision algorithms in computer graphics, and are connected through the use of self-

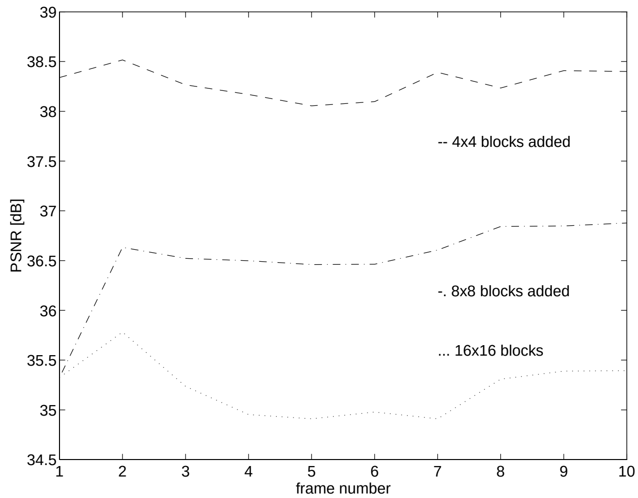

Figure 6: Coding results for the first ten frames of "Miss America": distortion evaluation.

similarity. The complexity of the encoding process which usually requires exhaustive search in the parameter space has not been addressed. There are faster algorithms that sacrifice performance for speed [18]. Complexity of the decoder was shown to be comparable to that of a DCT based method [4]. Multiscale fractal image coding may be a candidate for telecommunications applications, such as image banks, that require progressive image transmission, or multimedia video-conferencing with different levels of service quality.

The main contribution of this work is the multiscale pyramid image coding algorithm and the connection to the two-scale difference equation and wavelet theory. The video coding approach we present is also original. There is a large number of questions regarding fractal coding algorithms still unanswered and we mention only those in our immediate attention range: the use of a different class of self-similarity transformations T eventually nonlinear, overlapping range blocks, preprocessing the affine term for better self-similarity match, search for a fast encoding algorithm, the use of partial self-similarity to generate wavelets and multiresolution analyses.

References

[1] A. E. Jacquin, *A Fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding.* PhD thesis, Georgia Institute of Technology, Aug. 1989.

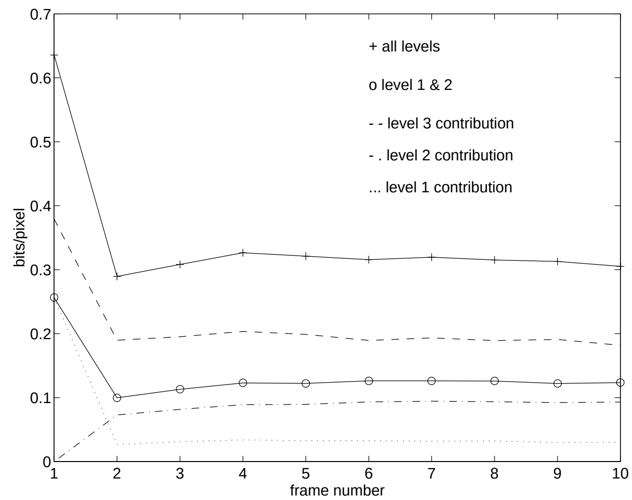

Figure 7: Coding results for the first ten frames of "Miss America": bit rates.

[2] 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.

[3] E. Zeidler, *Nonlinear Functional Analysis and its Applications*, vol. 1. Springer-Verlag, 1986.

[4] Y. Fisher, E. Jacobs, and R. Boss, "Fractal image compression using iterated transforms," in *Image and text compression* (J. A. Storer, ed.), ch. 2, pp. 35–61, Kluwer Academic, 1992.

[5] J. M. Beaumont, "Image data compression using data techniques," *BT Technol J*, vol. 9, pp. 93–109, Oct. 1991.

[6] M. F. Barnsley and L. Hurd, *Fractal Image Compression.* Wellesley, MA 02181: AK Peters, 1993.

[7] I. Daubechies and J. Lagarias, "Two-scale difference equations I. Existence and global regularity of solutions.," *SIAM J. Math. Anal.*, vol. 22, pp. 1388–1410, Sept. 1991.

[8] A. S. Cavaretta, W. Dahmen, and C. A. Micchelli, "Stationary subdivision," *Mem. Amer. Math. Soc.*, vol. 93, no. 453, 1991.

[9] W. Madich, "Some elementary properties of multiresolution analyses of $l^2(\Re^n)$," in *Wavelets-a tutorial in theory and applications* (C. Chui, ed.), pp. 259–294, Academic Press Inc., 1992.

[10] Y. Meyer, *Wavelets: algorithms and applications.* SIAM, 1993.

[11] D. S. Watkins, "Some perspectives on the eigenvalue problem," *SIAM Review*, vol. 35, pp. 430–471, Sept. 1993.

[12] P. J. Burt and E. H. Andelson, "The laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. COM-31, pp. 532–540, Apr. 1983.

[13] M. F. Barnsley, *Fractals Everywhere.* San Diego, CA.: Academic Press, 1988.

[14] A. E. Jacquin, "Fractal image coding: a review," *Proceedings of the IEEE*, vol. 81, pp. 1451–1466, Oct. 1993.

[15] W. Briggs, *A multigrid tutorial.* Philadelphia, Pa.: SIAM, 1987.

[16] J. Xu, "Iterative methods by space decomposition and subspace correction," *SIAM Review*, vol. 34, pp. 581–613, Dec. 1992.

[17] B. Hürtgen and P. Büttgen, "Fractal approach to low-rate video coding," *SPIE Visual Communications and Image Processing*, vol. 2094, pp. 120–131, Nov. 1993.

[18] A. Bogdan and H. E. Meadows, "Kohonen neural network for image coding based on Iteration Transformation Theory," *SPIE Neural and Stochastic Methods in Image and Signal Processing*, vol. 1766, pp. 425–436, July 1992.

[19] P. Lancaster and M. Tismenetsky, *The theory of matrices.* Academic Press, 2 ed., 1985.

[20] L. Kantorovich and G. Akilov, *Functional Analysis.* New York: Pergamon Press, 2nd ed., 1982.

[21] H. Stark, ed., *Image Recovery: Theory and Application.* London: Academic Press, 1987.

[22] K. Falconer, *Fractal Geometry: mathematical foundations and applications.* John Wiley & Sons, 1990.

[23] H. Otto Peitgen, H. Jürgens, and D. Saupe, *Chaos and Fractals; new frontiers of science.* New York: Springer-Verlag, 1992.

[24] N. Dyn, D. Levin, and J. Gregory, "A 4-point interpolatory subdivision scheme for curve design," *Comput. Aided Geom. Design*, vol. 4, pp. 257–268, 1987.

[25] M. Berger and Y. Wang, "Multidimensional two-scale dilation equations," in *Wavelets-a tutorial in theory and applications* (C. Chui, ed.), pp. 295–323, Academic Press Inc., 1992.

[26] A. Akanasu and F. Wadas, "On lapped orthogonal transforms," *IEEE Trans. on Signal Process.*, vol. 40, pp. 439–442, Feb. 1992.

[27] M. Wickerhause, "High-resolution still picture compression," *Digital Signal Processing, accepted*, 1992.