THE FRACTAL PYRAMID WITH APPLICATIONS TO IMAGE CODING

Alexandru Bogdan

Center for Telecommunications Research
Columbia University, New York, N.Y. 10027
Proc. of the IEEE Conf. ICASSP-95, May 1995, Detroit, MI.

ABSTRACT

We extend the iterated transformation theory (ITT) fractal image coding algorithm proposed by A. Jacquin [1] to generate a pyramid image representation. An $ITT-coded$ image is modeled as the solution of a second kind functional equation. This representation is iterated to form an $ITT-chain$ of functional equations which can serve as the framework for a multiscale signal decomposition. This formalism can be extended to accommodate hybrid ITT representations and, in the limit, $ITT-coded$ signals as a solution of a homogeneous functional equation. Existence of the $ITT-chain$ signal representation is shown to be connected to the eigen-structure of the linear operators of the associated functional equations. At each level of the $ITT-chain$ representation, the signal is decomposed into two parts which are not orthogonal. We use this decomposition to build an $ITT-pyramid$ representation for gray-tone images as well as for RGB color images.

1. INTRODUCTION

The *fractal image compression* $ITT-coding$ algorithm proposed by A. Jacquin [1] can be seen as an extension to automated image coding of the work of M. Barnsley [2] on iterated function systems (IFS). Several groups have worked to improve this method, mostly in reducing the encoding complexity [3]. Fractal coding is known for high compression and natural image appearance at very low bit rates and can be the preferred choice in some image applications. The algorithm has been used for color or video coding, in the form of individually compressed components/frames.

The ITT signal representation is usually explained using fixed-point theory [1]. A lossy $ITT-coded$ image $\bar{f}$ is the unique fixed-point of a contractive operator T in $\mathcal{I}$, the metric space of images: $\bar{f} = T\bar{f}: \ \bar{f} \in \mathcal{I}$. The operator T has a special structure which can be described as an affine transformation in $\mathcal{I}$, defined blockwise. The linear part of T maps blocks from two different scale representations of the signal, to exploit the self-similarity present in many natural phenomena. In image coding, the operator T is the code for the original image $\mathbf{f}$. Encoding $\mathbf{f}$ means finding an operator T having a fixed-point $\bar{f} \approx \mathbf{f}$, while decoding is equivalent to finding the fixed-point $\bar{f}$ by iterating T starting with an image selected at random. When the self-similarity of the signal is well modeled by the linear part of T, the description of T is much shorter than that of $\mathbf{f}$, resulting in high signal compression.

Preserving the affine structure of the operator T we formulate the ITT signal representation in terms of a two-scale functional equation of the second kind [4]. This representation can be iterated to obtain an $ITT-chain$ of functional equations which is used to build coding algorithms. We investigate the general properties of the $ITT-chain$ signal representation and connections to other methods. As an application example, we present a pyramid image coding algorithm for gray and color images.

This work was supported in part by the National Science Foundation under Grant CDR-88-11111.

2. THE $ITT-CHAIN$ SIGNAL REPRESENTATION

Images are usually modeled as a real function of two variables with compact support. The case of interest in applications is that of image representations in digital computers, such that an image f belongs to a *Hilbert* or, in general, a metric space $\mathcal{I}$ which is discrete and finite dimensional.

2.1. The ITT algorithm

The ITT representation [1] described briefly in section 1 can be written in the form of a two-scale functional equation:

$$f(x) = Tf(x) = U_L f(x) + b = A_L f(2x) + b \qquad (1)$$

in a metric space, where x is the space coordinate, A_L, U_L are linear operators and $f, b \in \mathcal{I}$. Coding f is equivalent to finding equation (1) parameterized by $\{A_L, b\}$, while decoding means solving the equation, usually using an iterative algorithm. In a digital coder A_L is a very sparse matrix and has a special structure generated by the well known self-similar block encoding algorithm. The free term b represents the gray levels offset at each range block in the original fractal compression algorithm [1] and can be seen as an image with constant gray levels at each range block. The uniqueness of the decoded image $\bar{f}$ depends on the spectral structure of the linear operator A_L [4]. We have two cases, depending if $b = \mathbf{0}$ or not, where $\mathbf{0} \in \mathcal{I}$ is the image with all entries equal to zero.

In the case $b \neq \mathbf{0}$, equation (1) has an unique solution

$$\bar{f} = (I - U_L)^{-1} b, \tag{2}$$

if no eigenvalue of U_L is equal to 1. When U_L is a contractive operator, equation (1) can be solved using the successive approximations method $f_{n+1}(x) = A_L f_n(2x) + b$, which is also the preferred method in ITT decoding. The contractivity condition $||A_L|| < 1$ depends on the metric which defines the space $\mathcal{I}$. A more general result holds [4] and is connected to the eigen-structure of the linear operator. The iterative algorithm can be used when the spectral radius of the linear operator is less than one: $\rho(U_L) < 1$. Then, the successive approximations method (indexed by n) will converge for any choice of the initial conditions f_0 and any b. When $\rho(U_L) > 1$, we have to obtain the solution directly using (2). In a practical digital image coding algorithm where $\mathcal{I}$ is finite dimensional, decoding the ITT representation (1) is equivalent to solving a large sparse system of linear equations. This type of problems are studied in numerical analysis and solutions are usually obtained using iterative algorithms such as Jacobi or Gauss-Seidel [5].

In the special case $b = \mathbf{0}$ the formal solution (2) does not exist. A solution of the homogeneous linear equation $f(x) = U_L f(x)$ can be seen as an eigenfunction associated with the eigenvalue $\lambda = 1$. It is possible to use the homogeneous equation for ITT signal representation. Decoding in this case is equivalent to solving an eigenvalue problem, which is well known in numerical analysis [6]. When a linear operator has a dominant eigenvalue with multiplicity one, the associated eigenvector is uniquely defined except for a multiplicative constant. From the various algorithms used to recover the dominant eigenvector, we describe the *power method* which is probably the simplest and best known.

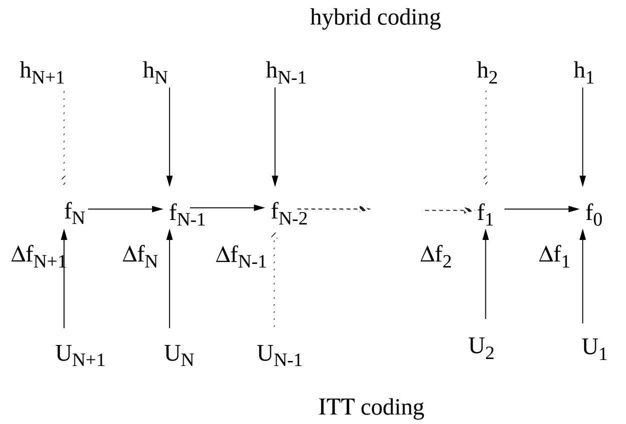

Figure 1: Hybrid ITT pyramid signal representation. A dotted arrow means that the component is not present at that pyramid level.

We use the iterative method

$$f_{n+1}(x) = k_n A_L f_n, \; n = 0, 1, \ldots, \tag{3}$$

where k_n is a renormalization factor. Assuming $\lambda_1 = 1$ is simple, $k_n = 1$ and f_0 is not orthogonal to the eigen-subspace of λ_1, the sequence $\{f_n\}$ converges to the dominant eigenvector. In practice $\lambda_1 \approx 1$ and we have to normalize at each iteration. An example of a signal defined by a linear operator is the scaling function in wavelet theory [7]. The two-scale difference equation is a homogeneous linear functional equation having a simple dominant eigenvalue equal to one and the mother wavelet (scaling function) is the associated eigenfunction.

2.2. Signal representation using the $ITT-chain$

We can iterate representation (1) to obtain a chain of functional equations:

$$\begin{aligned} f_0(x) &= A_L^1 f_0(2x) + f_1(x) \\ f_1(x) &= A_L^2 f_1(2x) + f_2(x) \\ &\ldots \\ f_{N-1}(x) &= A_L^N f_{N-1}(2x) + f_N(x). \end{aligned} \tag{4}$$

Assuming we know f_n, A_L^k $(k = 1, 2, \ldots, n-1)$ and that each equation is solvable, we can find f_0 and the intermediate solutions f_k. A pictorial representation of the flow of information in (4) is given in the lower part of Fig. 1. It looks very similar to the synthesis chain in a pyramid signal representation [8]. In image coding, if

$\bar{f}_0$ is the high detail image, and f_N is the coarsest approximation at level N, we add at each level r the detail information Δf_r which is coded in the linear operator U_r. We can rewrite the $ITT-chain$ decomposition (4) in a form similar to equation (2)

$$f_0 = (I - U_1)^{-1}(I - U_2)^{-1} \cdots (I - U_N)^{-1} f_N, \quad (5)$$

where I is the identity operator. Clearly the top of the pyramid f_N has to be obtained using a different representation. An option is the eigenfunction approach described in the previous section, which in Fig. 1 is determined by U_{N+1}. Another choice is a representation using a different algorithm which is shown as h_{N+1} in the same figure.

2.2.1. Hybrid representations

At each level r in the $ITT - chain$ we can add to the free term an element $h_r \in \mathcal{I}$ such that we have an ITT representation of the form (1). We obtain then a hybrid $ITT-chain$ which is pictured in Fig. 1. A hybrid term h_r is present or absent at a particular level, as dictated by the encoding algorithm.

2.3. Signal decomposition

The $ITT - chain$ representation (4) can be seen as a decomposition of the signal into several layers which are not orthogonal. For illustration we look at the representation (1) in the finite dimensional space $\mathcal{I}$. In applications, most of the eigenvalues of U_L are zero. Assume that the nonzero eigenvalues λ_i are distinct. Let e_i be the eigenvector associated with λ_i and assume we can complete the collection $\{e_i\}$ with the vectors $\{g_i\}$ such that $\{e_i, g_i\}$ is an orthonormal basis in $\mathcal{I}$. We expand $f, b \in \mathcal{I}$ as $f = \sum c_i e_i + \sum d_j g_j$ and $b = \sum \alpha e_i + \sum \beta_j g_j$. After substituting in (4) and some algebra we obtain

$$\sum \frac{\alpha_i}{1-\lambda_i} e_i + \sum \beta_j g_j = \sum \frac{\lambda_i \alpha_i}{1-\lambda_i} e_i + \sum \alpha_i e_i + \sum \beta_j g_j. \quad (6)$$

The left side of the equality sign in (6) is $\bar{f}$, the solution of the ITT equation. We see that spectral components which are not present in U_L are passed unchanged from the free term b to the solution $\bar{f}$. When $\lambda_i \neq 0$ there is a contribution from the term $\frac{\lambda_i \alpha_i}{1-\lambda_i} e_i$, determined by U_L. This contribution is significant when λ_i is close to one, since in the limit, the coefficient of e_i grows without bound as $\lambda_i \to 1$. To have an orthogonal decomposition b and $U_L f$ must be orthogonal or b must be orthogonal to the subspace spanned by $\{e_i\}$. In this case, the only possible solution has $\alpha_i = 0$ for all i. Another possibility is that U_L is a projection operator which means that λ_i can be only 0 or 1. This case is also ruled out because we loose the uniqueness of the representation $\bar{f}$.

level/ block size	bit rate	PSNR
5. 32x32 pels	.0236 bpp	22.38 dB
4. 16x16 pels	.0956 bpp	25.9 dB
3. 8x8 pels	.41 bpp	29.02 dB
2. 4x4 pels	.81 bpp	33.08 dB
1. 2x2 pels	1.02 bpp	33.51 dB

Table 1: Pyramid fractal coding of "Lena" (512×512 pixels).

3. APPLICATIONS TO IMAGE CODING

The $ITT - chain$ representation (4) can be seen as a predictive coder. At each level r, f_r is composed of the term f_{r+1} obtained from the previous level and additional information stored in U_l. In image applications we can run the $ITT-chain$ in various ways. For interframe video coding the index r is the time or frame number [9]. In this work r indexes the scale, to generate a pyramid image representation.

3.1. Pyramid image coding

In a pyramid gray image coder, we use at each level r, a tiling of the image support with equal size ($2^r \times 2^r$ pixels) range blocks. At level $r = 0$, the size of a range block is 1 pixel and lossless reconstruction is possible. Each range block is coded with 26 bits at each level. If a good approximation is obtained from level $r+1$ then the range block is not encoded and only a flag bit is set.

In simulations we use 5 levels for $r = 2 \div 5$ for a lossy representation. The top of the pyramid f_5 (range blocks of size 32×32 pixels is encoded using the homogeneous equation method described in section (2.1). The fractal pyramid image representation combines in the same code the *progressive* and *hierarchical* modes described in the JPEG image compression standard. The method described above corresponds to the *progressive* mode where we have a coarse representation which is updated at each step, keeping the image size constant. It can be shown that the same code in a fractal representation can be decoded at an arbitrary scale and the same is true for the chain representation (4). In the *hierarchical* mode we increase the resolution when adding a new level of detail. In the fractal

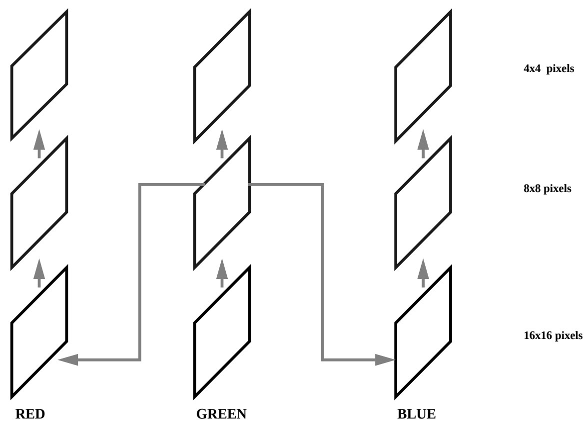

Figure 2: Multiscale pyramid used in the coding of RGB color images. It was found that using the second level of the Green image as the prediction for the other color components improves compression.

pyramid we have to redecode all the levels previously displayed which seems to be computationally intensive but it is not. Since decoding is an iterative process, using an upsampled version of a low resolution image as a starting image, reduces the number of iterations.

A sample coding of the well known "Lena" (no entropy coding) is presented in Table 1. The quantitative evaluation of the algorithm is in the same range as the JPEG standard at moderate bit rates and is better at very low bit rates. Visual quality is much better at low bit rates (it is well known that at low bit-rates PSNR is meaningless).

3.2. Color image coding

An example of an $ITT-chain$ image coding in the RGB colorspace is presented in Fig. 2. The chain representation is used in two ways. Each color component is coded using the still image pyramid representation presented in the previous subsection. The $GREEN$ component is individually coded and then it is used to predict the other color components. Various combinations for other color space representations are possible. Quantitative coding results are consistent with the $ITT-pyramid$ image simulations presented previously.

4. CONCLUSIONS

We have shown that the fractal image coding method proposed by A. Jacquin [1] can be extended using the framework of functional equation representations. The proposed $ITT-chain$ representation is used to build a multiscale pyramid coding algorithm for still gray and color images. We discuss the general properties of the representation and connections to other methods. Implementation in software show compression results comparable to other state-of-the-art methods.

There are many open problems in $ITT-coding$. Our formulation does not show how to build an efficient encoder, but can be used with any ITT encoding algorithm. We are currently investigating a different class of mappings U_l possibly nonlinear, optimal bit-allocation among the different levels of the $ITT-chain$ and applications to other areas such as pattern recognition.

5. REFERENCES

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

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

[3] 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.

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

[5] W. Hackbusch, *Iterative solution of large sparse systems of equations*. No. 95 in Applied Mathematical Sciences, Springer-Verlag, 1994.

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

[9] A. Bogdan, "Multiscale (inter/intra-frame) fractal video coding," *Proceedings of the IEEE Conf. ICIP-94*, vol. 1, pp. 760–764, Nov. 1994. Austin, Tx.