TREIZIÈME COLLOQUE GRETSI - JUAN-LES-PINS DU 16 AU 20 SEPTEMBRE 1991

Using Fractals to Model One-Dimensional Signals*

Monson H. Hayes, III, Greg Vines, and David S. Mazel

Georgia Tech Lorraine
2-3 Rue Marconi, 57070 Metz, France

RÉSUMÉ

Les Systèmes de Fonctions d'Itération (SFI) sont actuellement à l'étude comme une nouvelle approche pour modéliser le signal. Utilisant seulement quelques paramètres, un système de fonctions d'itération est capable de produire une large variété de fonctions compliquées, tel que les fonctions fractales, qui ont une propriété d'auto-ressemblance. Dans l'espoir d'obtenir des taux de compression de donnée très élevés dans la représentation du signal à temps discret, tel que la parole et l'image, la recherche s'oriente actuellement vers le problème inverse de la détermination des paramètres nécessaires a la SFI pour modéliser une fonction donnée. Dans cet article, nous décrivons les résultats récents obtenus par nous dans l'application du SFI à l'analyse et à la synthèse des signaux à temps discret.

ABSTRACT

Iterated Function Systems (IFSs) are currently being studied as a new approach to signal modeling. Using only a small number of parameters, an iterated function system is capable of producing a wide variety of complicated functions, such as *fractals*, which have a property of being self-similar. In the hopes of achieving large rates of data compression in the representation of discrete-time signals, such as speech and images, research efforts are currently being directed towards the *inverse problem* of determining the parameters needed by an IFS to model a given function. In this paper, we describe our recent research results in the application of iterated function systems to the analysis and synthesis of discrete-time signals.

1 Introduction

Iterated Function Systems (IFSs) have recently received a great deal of attention in the literature as a new technique for signal modeling. This interest stems from the fact that an IFS is simple in form and yet capable of producing complicated functions, many of which are fractal in form [1]. Iterated function systems have, in fact, been used to create images and other waveforms that closely resemble those found in nature [2]. Therefore, in the hope of achieving a large data compression rate in the representation of complex signals such as speech or audio waveforms and images, research is currently being directed towards the *inverse problem* of determining the parameters needed by an IFS to model a given signal or function [3-7]. In fact, recently published research results on the use of IFSs for image compression have been very promising [8-10]. This paper describes some of our recent work on the use of iterated function systems to model one-dimensional discrete-time signals.

2 Signal Modeling Using an IFS

In modeling a continuous-time signal, $x(t)$, or a discrete-time signal, $x(n)$, the goal is to reduce the amount of information or the number of parameters necessary to describe or represent the signal to within a given level of fidelity. For example, one approach that may be used to model a signal $x(t)$ is to select a set of signal samples, $x(t_i)$, and find an interpolation rule or function that will interpolate the signal between the sample points to produce a good approximation to the original signal. Suppose, for example, that $x(t)$ is a bandlimited signal having no energy above a frequency of Ω_0. Using a sampling period T_s with $T_s < \pi/\Omega_0$, the signal $x(t)$ may be reconstructed exactly from its samples $x(nT_s)$ using the interpolation formula

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT_s) \frac{\sin \Omega_0 (t - nT_s)}{\pi (t - nT_s)} \qquad (1)$$

However, since Equation (1) is restricted to bandlimited signals and assumes that the samples are obtained by uniform sampling, it may be advantageous to consider other methods for interpolating between sample values. In this paper we consider the use of interated function systems for interpolation [1,11].

2.1 Linear Fractal Interpolation Model

An IFS is a finite set of contraction mappings defined on a complete metric space $\mathcal{U}$ with a distance function ρ, i.e.,

$$\mathbf{w}_i : \mathcal{U} \rightarrow \mathcal{U} \qquad (2)$$

for $i = 1, 2, \ldots, M$ with

$$\rho[\mathbf{w}_i(\mathbf{x}), \mathbf{w}_i(\mathbf{y})] \leq s_i \cdot \rho[\mathbf{x}, \mathbf{y}] \qquad (3)$$

for all $\mathbf{x}, \mathbf{y} \in \mathcal{U}$ with $0 \leq s_i < 1$. Given a set of signal samples, $x_i = x(t_i)$, for $i = 0, 1, \ldots, M$, an iterated func-

*This work was supported in part by a grant from Rockwell International.

tion system may be used to interpolate $x(t)$ with a continuous, single-valued function that passes through the sample points, $x(t_i)$ using M affine maps of the form

$$\mathrm{w}_i \begin{pmatrix} t \\ x \end{pmatrix} = \begin{pmatrix} a_i & 0 \\ c_i & d_i \end{pmatrix} \begin{pmatrix} t \\ x \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix} \qquad (4)$$

for $i = 1, 2, \ldots, M$. Since map w_i will be a contraction if $|d_i| < 1$, d_i is called the *contraction factor* for the ith map. This parameter is independent of the sample points, $x(t_i)$, and is used to control the shape of the interpolation function between the sample points. The remaining parameters, a_i, c_i, e_i, and f_i, are constrained so that $x(t_0)$ and $x(t_M)$ are mapped by w_i to $x(t_{i-1})$ and $x(t_i)$:

$$\begin{aligned} \mathrm{w}_i : (t_1, x_1) &\longrightarrow (t_{i-1}, x_{i-1}) \\ \mathrm{w}_i : (t_M, x_M) &\longrightarrow (t_i, x_i). \end{aligned} \qquad (5)$$

Thus, for each i, the function w_i maps the signal $x(t)$ over the interval $[t_0, t_M]$ onto the interval $[t_i, t_{i+1}]$. As a result of the endpoint constraints, once the contraction factor for each map has been selected, the remaining parameters may be easily found [1,7].

In traditional approaches to signal modeling, such as AR and ARMA approximations, once the model parameters have been determined, the signal may be easily synthesized from the difference equation corresponding to the model. For an IFS, however, the problem of signal synthesis is more complicated since the signal is defined implicitly in terms of the M affine maps rather than in terms of a difference equation. Given the parameters of the affine maps w_i, the fractal interpolation function may be constructed using one of two algorithms – the *deterministic algorithm* or the *random iteration algorithm* [1].

2.2 Piecewise Self-Affine Fractal Model

The linear fractal interpolation model produces self-affine data using a set of affine functions that map the entire function into sections of itself. Self-affinity, however, is not necessarily a characteristic that a signal such as a speech utterance or an image might possess. A more general form of similarity, known as *piecewise self-affinity*, is a property in which *intervals* of the function are mapped to intervals. Specifically, given a set of sample points, $x(t_i)$ for $i = 0, 1, \ldots, M$, the interpolation function is constructed with M-affine maps of the form (4). However, in place of the end-point constraints for the self-affine model given by (5), for the piecewise self-affine a subset of the interval $[t_0, t_M]$ is mapped between each pair of sample points,

$$\begin{aligned} \mathrm{w}_i : (t'_{i-1}, x(t'_{i-1})) &\longrightarrow (t_{i-1}, x(t_{i-1})) \\ \mathrm{w}_i : (t'_i, x(t'_i)) &\longrightarrow (t_i, x(t_i)) \end{aligned} \qquad (6)$$

Thus, for each i, the function w_i maps the signal $x(t)$ over the interval $[t'_{i-1}, t'_i]$ onto the interval $[t_{i-1}, t_i]$. The interval $[t'_{i-1}, t'_i]$ is called the address interval because it can be thought of as the address of the source data for this map. The contraction factor of each map is again constrained to be real and bounded by one in magnitude, $|d_i| < 1$. In addition, the width of each address interval associated with a map is constrained to be strictly greater than the width of the section associated with the map

$$t'_i - t'_{i-1} > t_i - t_{i-1} \quad : i = 1, 2, \ldots, M. \qquad (7)$$

Once the contraction factors, addresses and interpolation points have been chosen, the remaining map parameters are easily determined from the end-point constraints (6). Synthesis of the piecewise self-affine fractal interpolation function may be accomplished in a manner similar to the deterministic algorithm with some slight modification [12,13].

3 Parameter Identification

Having described the self-affine and the piecewise self-affine fractal models, we now consider the problem of determining the parameters that are necessary to model a given signal or function. Here, we focus only on the self-affine fractal model, our results are easily extended to the piecewise self-affine model, which has been reported on elsewhere [7].

3.1 Self-Affine Fractal Models

One method that could be used to determine an appropriate set of interpolation points for a signal $x(n)$ would be to examine all possible combinations of interpolation points from the data. However, as the number of maps increases, this method rapidly becomes computationally infeasible. Some work has been done on parameter identification for the case in which the number of maps are fixed [5,6]. These methods, however, provide no flexibility in setting the accuracy of the model. More recently, another approach has been presented in which the number of maps is variable and the interpolation points and contraction factors necessary to model a signal to within a given degree of accuracy are found [7]. This algorithm is based on the fact that the resulting fractal function is self-affine. The inverse algorithm searches the given function for interpolation points and contraction factors that divide the function into sections that are affine transformations of the entire function. For a particular set of interpolation points, the contraction factors, d_i, of each map are determined and then the remaining map parameters are found using the endpoint constraints (4).

This algorithm was modified to increase the emphasis on reducing the number of maps used to model a given function and is provided in Table 1. Because of the somewhat arbitrary nature of the search method, the addition of starting alternately from the left and the right ends of the data was found to facilitate locating efficient maps. It was noted that small changes to the algorithm would often result in vastly different sets of maps - suggesting that the error surface is quite complex.

All the data was normalized prior to modeling in order to facilitate comparisons between different data sets. Good results were obtained with a minimum map width of 5 samples, and with the min_error constant set to 10^{-5}.

One important change from the original algorithm, which is not mentioned in the table due to a lack of space, was in step 7. If some maps have already been saved in step 8 and *either* no maps have been found yet for this scan *or* the best map found so far is not very good, then this could be the last map for the model if this map would cover all the remaining samples. In this case continue checking until the end of the data to be modeled is reached. This can result in the creation of overlapping maps, but usually it helps eliminate one map or more when the last map created would normally only cover a few points.

An Inverse Algorithm

1. Set scan direction - alternate left and right
   Set initial map to some minimum width
2. Calculate contraction factor d
3. If $|d| > 1$ go to 6
4. Calculate map parameters and error
5. If error < previous *or* error < min_error
   then save this map as best map
6. Advance end point of current map
7. If not at end of data to be mapped
   then go to 2
8. Save best map
   Set new range of data to be mapped
9. If there is more data to be mapped go to 1

Table 1: The Inverse Algorithm

3.2 Relaxing the Boundary Conditions

In the fractal interpolation methods covered so far, the IFS parameters have been calculated by forcing the model to meet the interpolation points exactly. These boundary conditions have been used to reduce the number of free variables, leaving only d_i, the *contraction factor*, to be calculated. By not taking advantage of the boundary conditions, we will increase the number of variables from one to three, but the model will be more flexible to fit the data. With the error defined as the difference in the modeled data, $x(n)$, and that predicted by the IFS, $w(n)$, a least squares minimization of

$$\xi = \sum_n [w(n) - x(n)]^2, \tag{8}$$

for each map over the range of n appropriate for each map, allows the parameters for a system of three equations to be determined.

This method has the advantage of not forcing the synthesized data to intercept the sampled data at any point exactly - instead we are making the synthesized data fit as well as possible *overall*. As a result, the overall error is not sacrificed in order to meet the interpolation points. The choice of interpolation points is somewhat arbitrary and they have been chosen in order to minimize the error with the constraint that the model actually touches these points. The existance of this constraint is not necessary and in judging the fidelity of the modeled data, one data sample typically is no more important than another in achieving a good fit. Therefore by relaxing this constraint, the performance of the model is improved. An example is shown in Figure 1 where there is a comparison between the various modeling techniques covered in this paper.

3.3 Nonlinear IFS

Much of the research to date has focused on the use of affine maps in IFSs [7,11]. The use of affine maps in an IFS results in a characteristic $1/f^{\beta}$ spectral shape, where β is determined by the map parameters [14]. This restriction to the type of data generated with an IFS may be allieviated through the use of non-affine maps. For the i^{th} map the j^{th} iterate of x and t can be written as:

$$x_{j+1} = c_i \cdot t_j + d_i \cdot x_j + f_i \tag{9}$$

$$t_{j+1} = a_i \cdot t_j + e_i \tag{10}$$

The transformation of t ensures that the mapped region of the original data, $[t_0, t_M]$, is mapped onto the desired destination region, $[t_i, t_{i-1}]$ in a one-to-one fashion, and is not changed with the addition of nonlinearities.

There are many possible nonlinearities to use in equation (9). A series of polynominals were tested, beginning by adding a quadratic term to the affine map (9) to get:

$$x_{j+1} = c_i \cdot t_j + d_i \cdot x_j + h_i \cdot x_j^2 + f_i. \tag{11}$$

As with the affine maps, the boundary conditions can be used to eliminate some of the variables, leaving two unknowns: d_i and h_i, which can be solved for using a least-squares minimization of the error as was defined in equation (8).

The check for a contraction mapping is vitally important for convergence of the IFS. When the map was affine, the contraction mapping test was a simple magnitude check on the value of the parameter d_i. With the addition of nonlinearities, the check becomes more complcated. With a simple quadratic in the map, the IFS is capable of having complex dynamics, including generating chaotic sequences.

As the order of the polynominal increases is was found that the chance of getting a contraction mapping decreases. With a cubic polynominal, the maps were rarely contractive and the best performance was obtained with a quadratic map including cross terms between x and t. Such a map is given by

$$x_{j+1} = c_i \cdot t_j + d_i \cdot x_j + g_i \cdot t_j^2 + h_i \cdot x_j^2 + k_i \cdot x_j \cdot t_j + f_i. \tag{12}$$

The method used in section 3.2 can also also be applied here and the boundary conditions need not be used. This again allows the synthesized waveform to fit closer to the desired shape, rather than pinning the function down at a few points.

One disadvantage to adding more terms to the map is that the addition of more terms increases the data storage requirement for each map. The coefficient for each nonlinear term must be saved along with the map parameters. Figure 1 illustrates the improvement offered by the quadratic map of equation(12).

4 Conclusion

We have seen several different IFS models for generating data, as well as several methods to determine the appropriate parameters to use. This is by no means a solved problem; as mentioned before, the search for the best set of maps involves a trade off between a computationally realistic algorithm and obtaining the optimum solution.

The relaxed affine method improves the performance of the interpolation models without adding any storage requirements. The addition of quadratic terms to the map helps the model's performance, however, the additional storage required for the coefficients of the quadratic can easily outstrip any performance gained through these extra terms. The goal here is to model the data efficiently, and keeping the maps simple, as well as keeping the number of maps low, is of paramount importance.

The area of fractal modeling with iterated function systems is an area that is full of promise. This paper has outlined several techniques for modeling one-dimensional signals including how to determine the parameters given a function or data set. We are continuing our efforts in this area and will report on our progress as it develops.

REFERENCES

1. Michael F. Barnsley, *Fractals Everywhere*, Academic Press, Inc. New York, 1988.
2. B. B. Mandelbrot, *The Fractal Geometry of Nature*, W. H. Freeman and Co., New York, 1982.
3. M. F. Barnsley, V. Ervin, D. Hardin and J. Lancaster, "Solution of an Inverse Problem for Fractals and Other Sets," *Proc. Natl. Acad. Sci. USA*, 83, 1986.
4. Arnaud Jacquin, *A Fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding*, Ph. D. Dissertation, Georgia Institute of Technology, 1989.
5. Giorgio Mantica and Alan Sloan, "Chaotic Optimization and the Construction of Fractals: Solution of an Inverse Problem" in *Complex Systems* (1989) vol. 3, pp. 37-62.
6. W. Douglas Withers, "Newton's Method for Fractal Approximation" in *Constructive Approximation* (1989) vol. 5, pp. 151-170.
7. D. S. Mazel and M. H. Hayes, "Fractal Modeling of Discrete-Time Signals", to appear in *IEEE Trans. on Signal Processing.*
8. Michael F. Barnsley and A. D. Sloan, "A Better Way to Compress Images," in *Byte*, Jan. 1988.
9. Elizabeth Corcoran, "Not Just a Pretty Face," in *Scientific American*, March 1990.
10. Glenn Zorpette, "Fractals: Not Just Another Pretty Picture," in *IEEE Spectrum*, October 1988.
11. Michael F. Barnsley, "Fractal Functions and Interpolation" in *Constructive Approximation* (1986) vol. 2, pp. 303-329.
12. Michael F. Barnsley and Arnaud E. Jacquin, "Application of Recurrent Iterated Function Systems to Images" in SPIE vol. 1001, Visual Communications and Image Processing, 1988.
13. Michael F. Barnsley, J. H. Elton and D. P. Hardin, "Recurrent Iterated Function Systems" in *Constructive Approximation* (1989) 5:3-31.
14. Alex P. Pentland, "Fractal-Based Description of Natural Scenes" in *IEEE Trans. on PAMI.* (1984) PAMI-6, pp. 661-674.

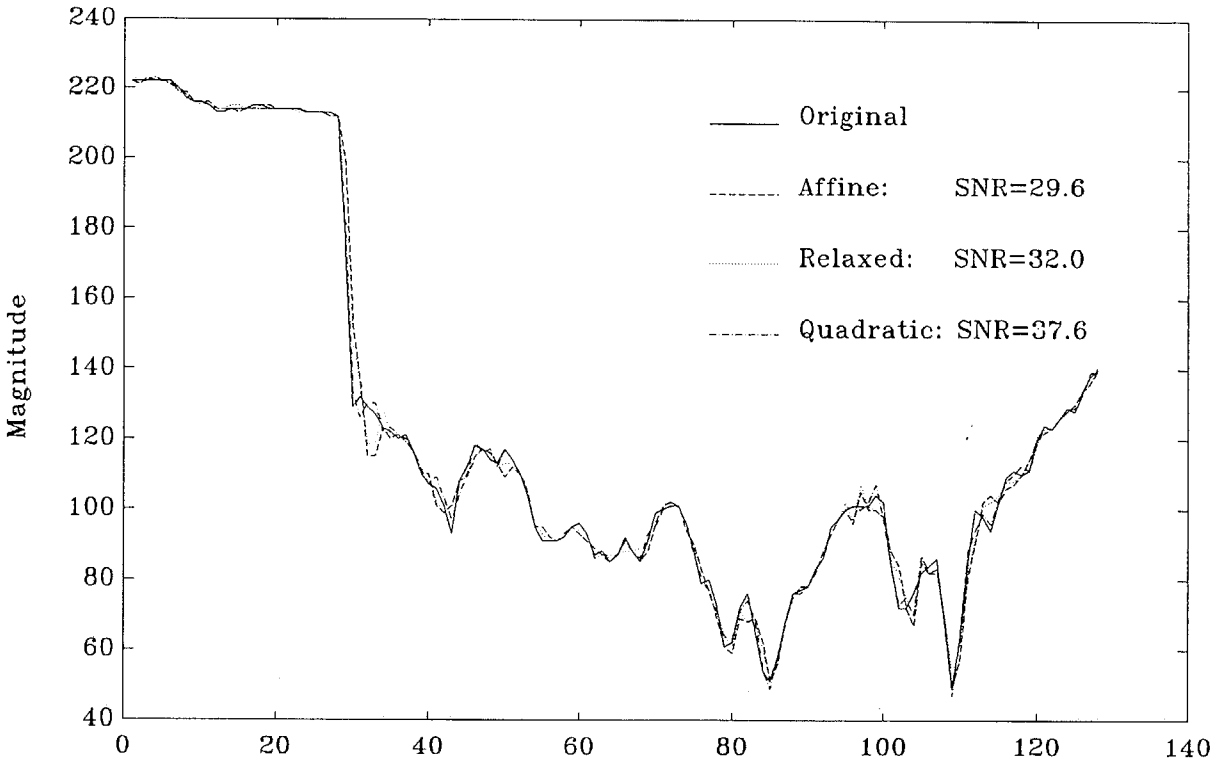

Figure 1. IFS Modeling Comparison