

Fractal Interpolation of Images and Volumes*

Jeffery R. Price[†] Monson H. Hayes III

Center for Signal and Image Processing
School of Electrical and Computer Engineering
Georgia Institute of Technology

price@ece.gatech.edu

Abstract

In this paper we present a method for constructing fractal interpolation surfaces and volumes through points sampled on rectangular lattices. Unlike other surface constructions ours uses rectangular rather than triangular tilings, halving the number of required parameters. This method is no more complex than previous constructions and yet does not suffer from their limitations. Additionally, our construction extends easily to volumetric interpolation, for which there were no previous (continuous) constructions. In addition to an example with synthetic data, a real image is interpolated using a fractal surface. Limitations and possible improvements are mentioned.

1. Introduction

Interpolation has long been an important tool for data visualization. Classical interpolation techniques fit elementary functions (e.g., lines and cubics) to given data points in order to render a connected visualization of the samples. Such elementary functions often imbue the visualization with a degree of smoothness that may not be consistent with the nature of the data. Fractals [1] and fractal interpolation [2] have been applied to prevent such inappropriate smoothing. Fractal interpolation functions (FIFs) have also been used to interpolate turbulent speech signals [3] as well as to model mountain profiles, seismic data, and electrocardiograms [4].

The motivation of this work was to explore the potential use of fractal interpolation in the context of image interpolation. In this respect, we consider points of an image to be samples (on a uniform rectangular grid) of a continuous

surface. As first presented in [5], the construction of (continuous) FIFs is relatively straightforward. Creating *continuous* fractal interpolation surfaces (FISs), and similarly fractal interpolation volumes (FIVs), however, is more difficult (excluding the tensor product cases).

In Section 2, we review the basics of FIFs and note two equivalent forms they may take – the latter of which facilitates our FIS and FIV constructions. In Section 3, we first state the FIS problem and describe the difficulties involved in constructing continuous FISs. We next briefly describe previously proposed constructions and point out their restrictions. We then detail our FIS construction and provide a synthetic example. In Section 4, we describe how the ideas behind our FIS construction can be extended to create continuous FIVs. In Section 5, we note that many fractal interpolation problems can be reduced to a simple matrix/vector expression that can, in turn, be used to aid in selecting free parameters. Finally, in Section 6, we present a portion of a fractal interpolated image and make some concluding remarks.

2. Fractal interpolation functions

For the classic linear FIF we have a set of data points

$$\{(x_n, y_n) \in D \times \mathbb{R} : n \in [0, 1, \dots, N]\}, \quad (1)$$

where x_n is strictly increasing and $D = [x_0, x_N] \subset \mathbb{R}$ is a closed interval. We seek a continuous function $f : D \rightarrow \mathbb{R}$ that interpolates this data according to

$$f(x_n) = y_n \quad \text{for } n \in [0, 1, \dots, N]. \quad (2)$$

Following the standard form used in the signal processing literature, such FIFs are constructed using N affine mappings of the form

$$w_n \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_n & 0 \\ b_n & \gamma_n \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} c_n \\ d_n \end{pmatrix} \quad \text{for } n \in [1, \dots, N], \quad (3)$$

*©1998 IEEE. In *Proceedings of the 32nd Asilomar Conference on Signals, Systems, and Computers*, vol. 2, pp. 1698-1702, November, 1998, Pacific Grove, CA, USA.

[†]J. Price is a Kodak Fellow. Funding for this research has been graciously provided by the Eastman Kodak Company.

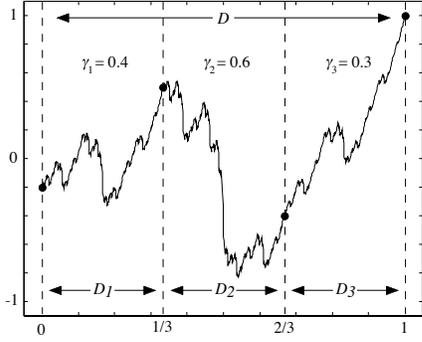


Figure 1. Example fractal interpolation function.

with the interval endpoint constraints

$$w_n \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} x_{n-1} \\ y_{n-1} \end{pmatrix} \quad \text{and} \quad w_n \begin{pmatrix} x_N \\ y_N \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} \quad \text{for } n \in [1, \dots, N]. \quad (4)$$

We refer to a_n as the *domain contraction factors* and γ_n simply as the *contraction factors*. Equations (3) and (4) imply that each map w_n horizontally “shrinks” (by a factor of a_n) and vertically scales (by a factor γ_n) the entire function over the interval D , and maps it to the piece of the function over the interval $D_n = [x_{n-1}, x_n]$. (See Fig. 1.)

It is easily shown that with each contraction factor γ_n a (fixed) free parameter, the remaining parameters of each map in (3) are uniquely specified by the constraints of (4). With each γ_n chosen such that $|\gamma_n| < 1$, the collection of affine mappings defined by (1)-(4) form a hyperbolic iterated function system (IFS). In other words, there exists a unique (nonempty) compact set $G \subset \mathbb{R}^2$ such that

$$G = \bigcup_{n=1}^N w_n(G). \quad (5)$$

Moreover, it can be shown that this set G is the graph of a continuous function $f : [x_0, x_N] \rightarrow \mathbb{R}$ that interpolates the data set according to (2).

Detailed in [5], but ignored in the signal processing literature until [6], is an equivalent form for the FIF associated with (1)-(4) that is expressed as follows:

$$w_n(x, y) = (L_n(x), F_n(x, y)) \quad (6a)$$

$$L_n(x) = a_n x + c_n \quad (6b)$$

$$F_n(x, y) = h(L_n(x)) + \gamma_n (f(x) - b(x)) \quad (6c)$$

where the *height function* $h(x)$ is the piecewise linear interpolation through the data points, and the *base function* $b(x)$ is the linear function through (x_0, y_0) and (x_N, y_N) .

In this alternative form, the subinterval endpoint constraints become

$$\begin{aligned} L_n(x_0) &= x_{n-1} \quad \text{and} \quad L_n(x_N) = x_n \\ F_n(x_0, y_0) &= y_{n-1} \quad \text{and} \quad F_n(x_N, y_N) = y_n. \end{aligned} \quad (7)$$

Note that $L_n(x)$ of (6b) describes the horizontal “shrinking” and mapping of D onto D_n . We will refer to these functions as the *domain contractions*. What has not been considered previously, but is necessary for our FIS and FIV constructions, is to allow the domain contraction factors a_n to be negative. A negative value for a_n implies that the entire function over D is reflected about its center point when mapped to D_n . (These reflections are similar to transforms used in fractal image compression.) This reflection just reverses the endpoint constraints of (4) and (7). This equivalent form for the FIF will be used in the next sections to aid in our FIS and FIV constructions.

3. Fractal interpolation surfaces

3.1. Problem Statement

For the surface interpolation problem, we begin with a data set that can be expressed similarly to (1) as

$$\{(\mathbf{x}_n, y_n) \in D \times \mathbb{R} : n \in \mathcal{N}\}, \quad (8)$$

where now $\mathbf{x}_n = (x_n^1, x_n^2)$, $D \subset \mathbb{R}^2$ is closed, and \mathcal{N} represents some ordering of the data set. In our case of a uniform rectangular lattice, we have

$$\mathcal{N} = [0, 1, \dots, N_1] \times [0, 1, \dots, N_2]. \quad (9)$$

To interpolate the data of (8) we seek a continuous surface $f : D \rightarrow \mathbb{R}$ such that

$$f(\mathbf{x}_n) = y_n \quad \text{for } n \in \mathcal{N}. \quad (10)$$

We attempt to construct such a surface using maps similar to (6), where x now becomes the 2-vector $\mathbf{x} \in D \subset \mathbb{R}^2$ and $L_n(\mathbf{x})$ can take more general forms (as noted later).

Like the FIF, the FIS construction requires each domain contraction $L_n(\mathbf{x})$ to take the entire domain $D \subset \mathbb{R}^2$ (or a subregion of D for the recurrent IFS) onto the smaller “subdomain” D_n , where the union of these subdomains covers D . In the FIF case, these subdomains were the subintervals we referred to as D_n . For the FIS, these subdomains are areas and can take various shapes as illustrated in Fig. 2. The intuitive extension of the FIF would imply that these subdomains in \mathbb{R}^2 should be rectangular. Such rectangular subdomains have, however, proved troublesome for constructing continuous FISs. In Section 3.3 we describe how to alleviate this problem, but mention here that previous FIS constructions have resorted to triangular subdomains.

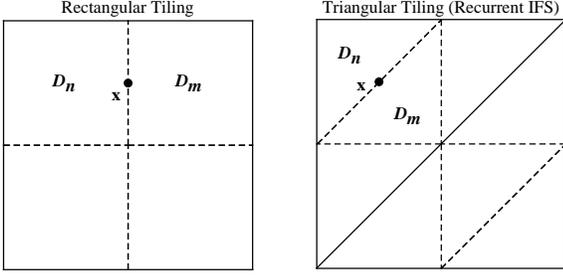


Figure 2. Domain for fractal interpolating surfaces over rectangular lattice, possible subdomain tilings, and subdomain boundary points.

The key difficulty in constructing FISs involves ensuring continuity. Adjacent subdomains are associated with different mappings, and yet share common points. To guarantee continuity, we require that these adjacent mappings produce the same values at these common points. Referring to Fig. 2, let \mathbf{x} be a point on the boundary of adjacent subdomains D_n and D_m , which are associated with mappings w_n and w_m , respectively. Note first that from (6) we can show that $F_n(\mathbf{x}, y) = f(L_n(\mathbf{x}))$, and therefore

$$f(\mathbf{x}) = h(\mathbf{x}) + \gamma_n [f(L_n^{-1}(\mathbf{x})) - b(L_n^{-1}(\mathbf{x}))] \quad \text{for } \mathbf{x} \in D_n. \quad (11)$$

For the boundary point we have both $\mathbf{x} \in D_n$ and $\mathbf{x} \in D_m$. With this in mind, we can use (11) to see that for continuity it is required that

$$\begin{aligned} \gamma_n [f(L_n^{-1}(\mathbf{x})) - b(L_n^{-1}(\mathbf{x}))] \\ = \gamma_m [f(L_m^{-1}(\mathbf{x})) - b(L_m^{-1}(\mathbf{x}))] \end{aligned} \quad (12)$$

for all \mathbf{x} on the boundaries of adjacent subdomains. These requirements are sometimes referred to as the “join-up” conditions [7]. In the FIF, the boundary between adjacent subdomains consists of only a single point. By the endpoint constraints of (4) or (7) the join-up conditions are automatically satisfied. For the FIS, however, adjacent subdomains share infinitely many points along boundaries. Applying constraints similar to those of the FIF construction will not work

3.2. Previous constructions

The construction of FISs has been investigated by several authors [2], [7]-[9]. Here we briefly review these constructions, and point out some of their inherent restrictions.

First, we note that all of the previous constructions employ triangular subdomains. For data over rectangular lattices, such a triangular tiling requires twice as many map-

pings as a rectangular tiling, since each potential rectangular subdomain must be divided into two triangles. Additionally, we note that using such a triangular tiling when D is in fact rectangular requires the recurrent IFS formalism. In this case, larger triangular subregions of D map to the smaller triangular subdomains [7],[9]. For an illustration of these ideas refer to Fig. 2.

The first published construction of a continuous FIS was described in [8]. This construction requires all the boundary data of D to be coplanar. As noted in [7] this will produce an FIS with many straight line segments, potentially a significant drawback in visualizing many phenomena. The coplanar boundary requirement is removed in the FIS construction of [7], but only in the special case where every mapping has the same contraction factor. This requirement can prove limiting as well since it implies that the surface is equally “rough” all over, and precludes selecting the contraction factors to satisfy any appropriate criteria. In [2] the author increases the dimension of the affine mappings from (3) in order to construct FISs (and FIVs). Although unmentioned, this construction either produces discontinuous surfaces (and volumes) or reduces to the case where the contraction factors must be constant.

The breakthrough that permits our construction was realized by Zhao in [9]. Using a form similar to (6), Zhao allows the contraction factors γ_n to become a continuous “contraction function” $\gamma(\mathbf{x})$. When expressed similarly to (6c), this yields

$$F_n(\mathbf{x}, y) = h(L_n(\mathbf{x})) + \gamma(\mathbf{x})(f(\mathbf{x}) - b(\mathbf{x})). \quad (13)$$

With $\gamma(\mathbf{x})$ now a continuous function, the join-up conditions of (12) reduce to

$$\begin{aligned} f(L_n^{-1}(\mathbf{x})) - b(L_n^{-1}(\mathbf{x})) \\ = f(L_m^{-1}(\mathbf{x})) - b(L_m^{-1}(\mathbf{x})). \end{aligned} \quad (14)$$

The simplest method to satisfy these join-up conditions is to ensure that

$$L_n^{-1}(\mathbf{x}) = L_m^{-1}(\mathbf{x}) \quad (15)$$

for every point \mathbf{x} on the boundaries of adjacent subdomains D_m and D_n . This means that L_n and L_m should map the same points (from an edge) of D onto the common boundary (edge) points shared by D_m and D_n . To meet this condition, Zhao employs triangular subdomains, and the domain contractions L_n are chosen to rotate D (or a triangular subregion) appropriately.

3.3. Rectangular subdomain FIS

Here we describe our (in fact quite simple) extension of the approach in [9] that allows the use of rectangular subdomains. The central problem to solve is satisfying the join-up

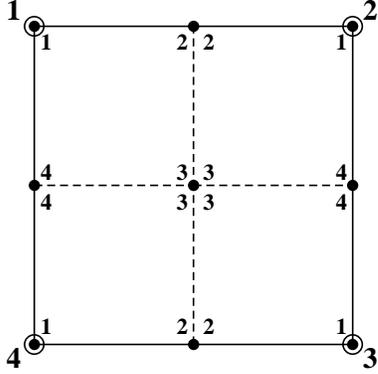


Figure 3. Rectangular domain contractions to satisfy join-up conditions.

conditions of (15). First note that domain contractions that employ only rotations of (a rectangular) D cannot satisfy (15). The key to solving this problem is realizing that, contrary to the triangular case, rotations alone do not describe all of the domain contraction possibilities for a rectangular D . As mentioned at the end of Section 2, we should also consider reflections. Explicitly adapting the notation of (6b) for the rectangular FIS, the domain contractions can be written as

$$L_n(\mathbf{x}) = \begin{pmatrix} a_n^1 & 0 \\ 0 & a_n^2 \end{pmatrix} \mathbf{x} + \begin{pmatrix} c_n^1 \\ c_n^2 \end{pmatrix}. \quad (16)$$

The four possible sign combinations of a_n^1 and a_n^2 realize four different combinations of reflections about the two \mathbf{x} axes. Using these combinations appropriately, one can construct an FIS over rectangular subdomains such that the join-up conditions of (15) are indeed satisfied. An illustration of this is shown in Fig. 3, where the large numbers, circled points, and solid lines indicate the vertices and connecting edges of D . The smaller numbers in the subdomains (indicated by dashed lines) indicate where these vertices are mapped by the domain contractions. For larger data sets, this pattern is repeated as necessary. Note that each of the four possible reflection combinations is necessary to satisfy the join-up conditions. With the domain contractions chosen in this fashion, we now turn our attention to the selection of the contraction function $\gamma(\mathbf{x})$.

Any continuous function for $\gamma(\mathbf{x})$ is suitable. In order to preserve the simplicity of the FIF, however, it might be desirable to choose a function that can be represented with a number of parameters close to, or perhaps even much less than, the number of maps. This would allow the same flexibility of the FIF without an increase in complexity. We have chosen to use a piecewise linear form (i.e. bilinear) for $\gamma(\mathbf{x})$ where there is a parameter γ_n associated with each data point (rather than with each map). In other words, we have a map for each $n \in [1, \dots, N_1] \times [1, \dots, N_2]$ but a con-

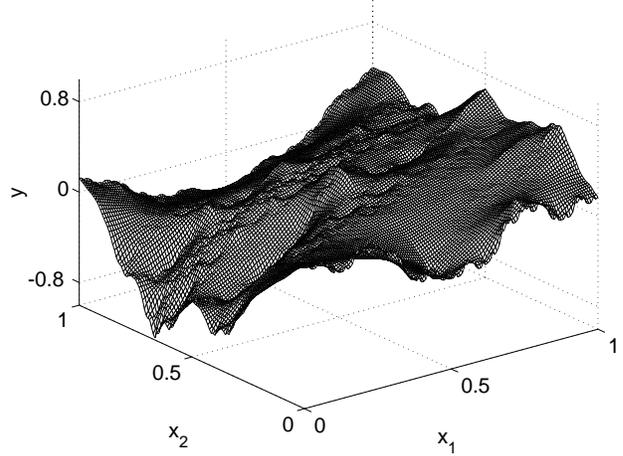


Figure 4. Example fractal interpolation surface.

traction factor for each $n \in [0, 1, \dots, N_1] \times [0, 1, \dots, N_2]$. Additional points of the contraction function $\gamma(\mathbf{x})$ are calculated by simple bilinear interpolation of these contraction factors.

An example FIS through a 3×3 lattice of synthetic data points is shown in Fig. 4. The data points were assumed to be sampled uniformly on $D = [0, 1] \times [0, 1]$. The contraction function used was bilinear, and parameterized by a 3×3 array of contraction factors. The surface of Fig. 4 is composed of 129×129 points interpolated from the original 3×3 lattice.

4. Fractal interpolation volumes

The extension of our FIS construction to the volume case is straightforward. An equation similar to (16) is used, now in \mathbb{R}^3 . With three different a_n parameters, there are eight possible sign combinations. These correspond to the eight combinations of reflections across each of the three \mathbf{x} planes. In the volumetric case, the domain contractions now map larger parallelepipeds to smaller parallelepipeds. Using appropriate reflections, similar to Section 3.3, a “rectangular” tiling can be constructed such that common points on adjacent “faces” (rather than line segments in the FIS case) of subdomains come from the same points on the face of the entire domain. In conjunction with a continuous (3-D) contraction function $\gamma(\mathbf{x})$, the join-up conditions will be satisfied, resulting in a continuous FIV.

5. Contraction factor selection

For many interpolation problems, only a few additional points between each data point are sought. In the FIF case, (11) can be used to reduce such a problem to the following

matrix/vector form

$$\mathbf{f} = \mathbf{h} + \mathbf{Q}\boldsymbol{\gamma}, \quad (17)$$

where the vectors \mathbf{f} and \mathbf{h} denote points of the FIF $f(x)$ and the height function $h(x)$, respectively [6]. The vector $\boldsymbol{\gamma}$ is composed of the contraction factors, and \mathbf{Q} is a (sparse) matrix whose entries are determined by only a few values of the difference function $(f(\mathbf{x}) - b(\mathbf{x}))$.

Employing the bilinear (or similarly parameterized) form for $\boldsymbol{\gamma}(\mathbf{x})$, as mentioned in Section 3.3, we can derive an expression (with appropriate dimensionality adjustments and data ordering) for the FIS and FIV cases:

$$\mathbf{f} = \mathbf{h} + \mathbf{Q}\mathbf{B}\boldsymbol{\gamma}. \quad (18)$$

The additional matrix \mathbf{B} represents the bilinear interpolation (or parameterized computation) of the contraction factors $\boldsymbol{\gamma}$ in order to calculate the necessary points of the contraction function $\boldsymbol{\gamma}(\mathbf{x})$. In the form of (18), various techniques (e.g., constrained optimization) can be used to find $\boldsymbol{\gamma}$ such that the interpolation \mathbf{f} possesses some quantifiable properties related to the known data points.

6. Real image example and conclusions

Unfortunately, experiments using our rectangular subdomain FIS for image interpolation have been largely unsuccessful. An exaggerated example is shown in Fig. 5, where a 21×21 section of the *Lena* image has been interpolated by a factor of 20. The rectangular tiling is evident in the grid-like structure of the image. Although our experiments indicate limited applicability for image interpolation, the flexibility of our FIS and FIV constructions may find use in other visualization endeavors. Additionally we note that careful selection of the contraction factors, using (18), may also prove beneficial

In summary, we have derived a method for constructing continuous fractal interpolation surfaces for data on rectangular lattices. Unlike prior constructions, ours allows the use of true rectangular subdomains and, additionally, does not suffer from the other constraints of these previous constructions. We also mentioned how our surface construction extends easily to the volume interpolation problem, for which there were no previous (continuous) constructions. Finally we noted that in many cases fractal interpolation problems can be expressed in a matrix/vector expression that allows the contraction factors to be selected according to some appropriate criteria.

References

[1] T. Lundahl, W. Ohley, W. Kuklinski, D. Williams, H. Gewirtz, and A. Most, "Analysis and interpolation

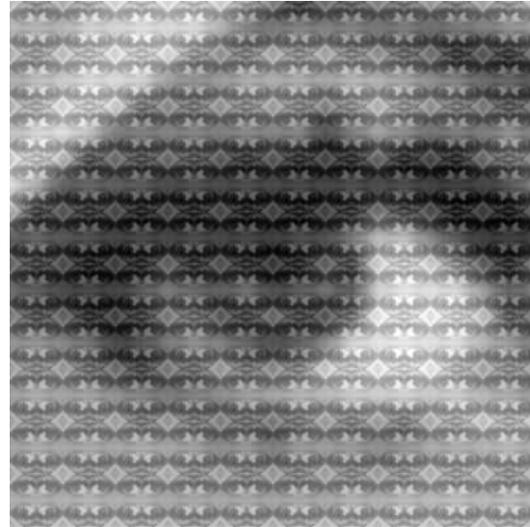


Figure 5. Poor image interpolation with FIS.

of angiographic images by use of fractals," in *Computers in Cardiology*, pp. 355–358, 1985.

- [2] C. M. Wittenbrink, "IFS fractal interpolation for 2D and 3D visualization," in *Proc. of the IEEE Visualization Conference*, pp. 77–84, 1995.
- [3] P. Maragos, "Fractal aspects of speech signals: Dimension and interpolation," in *ICASSP*, vol. 1, pp. 417–420, 1991.
- [4] D. S. Mazel and M. H. Hayes III, "Using iterated function systems to model discrete sequences," *IEEE Trans. on Signal Processing*, vol. 40, no. 7, pp. 1724–1734, 1992.
- [5] M. Barnsley, "Fractal functions and interpolation," *Constructive Approximation*, vol. 2, pp. 303–329, 1986.
- [6] J. R. Price and M. H. Hayes III, "Resampling and reconstruction with fractal interpolation functions," *IEEE Signal Processing Letters*, vol. 5, no. 9, pp. 228–230, 1998.
- [7] J. Geronimo and D. Hardin, "Fractal interpolation surfaces and a related 2-D multiresolution analysis," *Journal of Mathematical Analysis and Applications*, vol. 176, no. 2, pp. 561–586, 1993.
- [8] P. R. Massopust, "Fractal surfaces," *Journal of Mathematical Analysis and Applications*, vol. 151, no. 1, pp. 275–290, 1990.
- [9] N. Zhao, "Construction and application of fractal interpolation surfaces," *The Visual Computer*, vol. 12, no. 3, pp. 132–146, 1996.