Elevation data compression using IFS-based three-dimensional fractal interpolation

Siyuan Wu (吴思源) and Yuanhua Zhou (周源华)

Institute of Image Communication & Information Processing, Shanghai Jiaotong University, Shanghai 200030

Received June 29, 2004

This letter presents a novel application of iterated function system (IFS) based three-dimensional (3D) fractal interpolation to compression elevation data. The parameters of contractive transformations are simplified by a concise fractal iteration form with geometric meaning. A local iteration algorithm is proposed, which can solve the non-separation problem when Collage Theorem is applied to find the appropriate fractal parameters. The elevation data compression is proved experimentally to be effective in reconstruction quality and time-saving.

OCIS codes: 100.3020, 100.6890, 280.0280.

Digital elevation model (DEM) file is a regularly spaced grid of remote sensing elevation points, the most common kind of which comes from United State Geological Survey (USGS). Since the DEM files are very large with high spatial resolution, it is of great significance to minimize their storage space. Conventional compression techniques, such as lossy formats (JPEG, Wavelet) and lossless formats (Huffman, Arithmetic Coding, and LZW), have been applied to elevation grids. However, the results turn out unsatisfactory.

Interpolation has been an important tool for terrain visualization and is actually a simple method for compression if the original terrain can be reconstructed from sampled one. However, the common linear or spline interpolation is unsuitable for DEM compression owing to their smoothness for high-frequency components. Fractal geometry[1] founded by Mandelbrot characterizes the self-similarity of nature, such as coastline and fractured surface of magnetic dielectric film[2]. Honda[3] proposed an algorithm of fractal interpolation for natural image, in which fractal dimension of an image is estimated and fractional brown motion (FBM) is generated. Although the method enjoys higher fidelity than the conventional interpolation, its limitations lie in the multifractal behavior of terrain and randomness of FBM.

This letter investigates three-dimensional (3D) deterministic fractal interpolation based on iterated function system (IFS) and the inverse problems of elevation data compression. IFS-based fractal interpolation was first described by Barnsley and has been used for various two-dimensional (2D) applications such as data visualization and signal processing[4]. However, the successful applications of 3D fractal interpolation are quite limited because it is hard to deal with the complicated form and boundary continuity[5–7].

A concise fractal iteration form introduced in Ref. [8] can generate continuous 3D fractal interpolation surface. Suppose rectangular domain $D = I \times J = \{(x, y), 4p_1 \le x \le p_2,\ q_1 \le y \le q_2\}$ and given original interpolation data set $\{(x_n, y_m, Z_{n,m}),\ p_1 = x_0 < x_1 < \cdots < x_N = p_2,\ q_1 = y_0 < y_1 < \cdots < y_M = q_2.$ for $n = 0, 1, \cdots, N,\ m = 0, 1, \cdots, M\}$. Let $I_n = [x_{n-1}, x_n]$, $J_m = [y_{m-1}, y_m]$, subdomain $D_{n,m} = I_n \times J_m$, where $n = 1, \cdots, N,\ m = 1, \cdots, M$. The contractive transformations of IFS $\{W_{nm}, n = 1, \cdots, N,\ m = 1, \cdots, M\}$ are defined as

$$W_{nm}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} L_n(x) \\ L_m(y) \\ F_{nm}(x, y, z) \end{bmatrix}, \tag{1}$$

where $L_n : I \to I_n$ and $L_m : J \to J_m$ are contractive homeomorphisms given by

$$L_n(x) = \begin{cases} x_{n-1} + \frac{x_n - x_{n-1}}{x_N - x_0}(x - x_0) & n\%2 = 1 \\ x_n + \frac{x_{n-1} - x_n}{x_N - x_0}(x - x_0) & n\%2 = 0 \end{cases}$$
$$L_m(y) = \begin{cases} y_{m-1} + \frac{y_m - y_{m-1}}{y_M - x_0}(y - y_0) & m\%2 = 1 \\ y_m + \frac{y_{m-1} - y_m}{y_M - y_0}(y - y_0) & m\%2 = 0 \end{cases}, \tag{2}$$

and $F_{nm} : R^3 \to R$ is a constant function defined on $D = I \times J$ as

$$F_{nm}(x, y, z) = g_{nm}[L_n(x), L_m(y)] + d[L_n(x), L_m(y)] \times [z - h(x, y)], \tag{3}$$

where $z = g_{nm}(x, y)$ is the bilinear interpolation function determined by the four points, $(x_{n-1}, y_{m-1}, z_{n-1,m-1})(x_n, y_{m-1}, z_{n,m-1})(x_{n-1}, y_m, z_{n-1,m})(x_n, y_m, z_{n,m})$; $z = h(x, y)$ is the bilinear interpolation function determined by the four points, $(x_0, y_0, z_{0,0})(x_0, y_M, z_{0,M})(x_N, y_0, z_{N,0})(x_N, y_M, z_{N,M})$; $z = d(x, y)$ is an arbitrary continuous function defined on $D = I \times J$ as $|d(x, y)| < 1$, which is called as the contraction function of fractal interpolation.

Fractal interpolation surface is the attractor of the above IFS. Since the parameters of contractive transformations are simplified by this concise fractal iteration form with explicit geometric meaning, the applications of 3D fractal interpolation become practical and convenient.

Elevation data compression is the inverse problem of IFS-based 3D fractal interpolation, which can find appropriate contraction function in Eq. (3) so that their attractors are approximate to the original data $\{P(x, y)\}$. The inverse problem of IFS-based fractal interpolation can be solved by using collage theorem[9].

An operator $T : C[I \times J] \to C[I \times J]$ can be deduced

from Eqs. (1)−(3)

$$Tg(x,y) = g_{nm}(x,y) + d(x,y) \times [g(L_n^{-1}(x), L_m^{-1}(y)) - h(L_n^{-1}(x), L_m^{-1}(y))] \quad \forall g \in C[I \times J] \qquad (x,y) \in I_n \times J_m. \tag{4}$$

When Collage theorem is applied to $p(x,y) \in C[I \times J]$, the above mentioned inverse problem can be rewritten as

$$\min E = \min \|p(x,y) - Tp(x,y)\|_2^2 = \min_{d(x,y)} \|p(x,y) - g_{nm}(x,y) - d(x,y) \times [p(L_n^{-1}(x), L_m^{-1}(y)) - h(L_n^{-1}(x), L_m^{-1}(y))]\|_2^2. \tag{5}$$

In this letter, the piecewise bilinear interpolation function is chosen as the contraction function $z = d(x,y)$, where the parameters $\{d_{nm}, n = 0,1,\cdots,N,\ m = 0,1,\cdots,M\}$ are associated with each interpolate points. The base function of bilinear interpolation function is given by

$$\phi_{nm} = \begin{cases} \frac{x_{n+1}-x}{x_{n+1}-x_n} \cdot \frac{y_{m+1}-y}{y_{m+1}-y_m} \\ \quad x \in [x_n, x_{n+1}], y \in [y_m, y_{m+1}] \\ \frac{x-x_{n-1}}{x_n-x_{n-1}} \cdot \frac{y_{m+1}-y}{y_{m+1}-y_m} \\ \quad x \in [x_{n-1}, x_n], y \in [y_m, y_{m+1}] \\ \frac{x-x_{n-1}}{x_n-x_{n-1}} \cdot \frac{y-y_{n-1}}{y_n-y_{n-1}} \\ \quad x \in [x_{n-1}, x_n], y \in [y_{m-1}, y_m] \\ \frac{x_{n+1}-x}{x_{n+1}-x_n} \cdot \frac{y-y_{n-1}}{y_n-y_{n-1}} \\ \quad x \in [x_n, x_{n+1}], y \in [y_{m-1}, y_m] \end{cases}$$

By replacing $d(x,y)$ in Eq. (5) with $d(x,y) = \sum_{n=0}^{N}\sum_{m=0}^{M} d_{mn} \cdot \phi_{mn}$, the inverse problem in Eq. (5) can be written as the nonlinear optimization for several variables $\{d_{nm}\}$

$$\min E = \min \|p(x,y) - Tp(x,y)\|_2^2 = \min_{\{d_{nm}\}} \sum_{nm} \|[p(x,y) - g_{nm}(x,y)] - (\sum_{uv} d_{uv} \cdot \phi_{uv}) \times [p(L_n^{-1}(x), L_m^{-1}(y)) - h(L_n^{-1}(x), L_m^{-1}(y))]\|_2^2. \tag{6}$$

Genetic algorithm is effective to solve the nonlinear optimization problem of several variables, however it is unrealistic for time-consuming computation when there are thousands of variables. When we attempt to gain the derivative equation set from the partial derivative for each variable, it is quite difficult to list the explicit form of equation set. Especially, the scale of equation set become huge for a large number of variables. Furthermore, the Collage distance depends on many variables $\{d_{nm}\}$ in each subdomain, which is different from the common block-based fractal coding[9]. Consequently, the equation set can not be simplified as separated and independent equation as that of block-based fractal coding[9], which leads to the hard-solving non-separation problem.

As far as the localization of bilinear interpolation function is concerned, a local iteration algorithm (LIA) is proposed to solve this non-separation problem, in which only the related four subdomains $\{D_{nm}, D_{n+1,m}, D_{n,m+1}, D_{n+1,m+1}\}$ are taken into auount the optimal value of d_{nm}

$$\min E_{nm} = \min_{d_{nm}} \sum_{u=n}^{n+1}\sum_{v=m}^{m+1} \| [p(x,y) - g_{uv}(x,y)] - (d_{u-1,v-1}\phi_{u-1,v-1} + d_{u-1,v}\phi_{u-1,v} + d_{u,v-1}\phi_{u,v-1} + d_{u,v}\phi_{u,v}) \times [p(L_u^{-1}(x), L_v^{-1}(y)) - h(L_u^{-1}(x), L_v^{-1}(y))] \|_2^2. \tag{7}$$

Figure 1 serves to demonstrate the algorithm. For DEM surface $\{(x,y,p(x,y)),\ x = 0,\cdots,K,\ y = 0,\cdots,K\}$ and given original interpolation data set with the regular sample rate of 1: S as $\{(x_n, y_m, P_{n,m})$, where $n = 0,1,\cdots,L, m = 0,1,\cdots,L, L = K/S\}$, let $\overrightarrow{A} \oplus \overrightarrow{B} = (a_{ij}) \oplus (b_{ij}) = \sum_{i=0}^{S}\sum_{j=0}^{S} (a_{ij} \cdot b_{ij})$, $\overrightarrow{A} \otimes \overrightarrow{B} = (a_{ij}) \otimes (b_{ij}) = (a_{ij} \cdot b_{ij}) = (c_{ij}) = \overrightarrow{C}$, E_{nm} in Eq. (7) can be rewritten as

$$\begin{aligned} E_{nm} = & \left\| \overrightarrow{\Delta_1} - (d_{n-1,m-1}\overrightarrow{I_1} + d_{n-1,m}\overrightarrow{I_2} + d_{n,m-1}\overrightarrow{I_3} + d_{n,m}\overrightarrow{I_4}) \otimes \overrightarrow{B_1} \right\|_2^2 \\ & + \left\| \overrightarrow{\Delta_2} - (d_{n,m-1}\overrightarrow{I_1} + d_{n+1,m-1}\overrightarrow{I_2} + d_{n,m}\overrightarrow{I_3} + d_{n+1,m}\overrightarrow{I_4}) \otimes \overrightarrow{B_2} \right\|_2^2 \\ & + \left\| \overrightarrow{\Delta_3} - (d_{n-1,m}\overrightarrow{I_1} + d_{n,m}\overrightarrow{I_2} + d_{n-1,m+1}\overrightarrow{I_3} + d_{n,m+1}\overrightarrow{I_4}) \otimes \overrightarrow{B_3} \right\|_2^2 \\ & + \left\| \overrightarrow{\Delta_4} - (d_{n,m}\overrightarrow{I_1} + d_{n+1,m}\overrightarrow{I_2} + d_{n,m+1}\overrightarrow{I_3} + d_{n+1,m+1}\overrightarrow{I_4}) \otimes \overrightarrow{B_4} \right\|_2^2, \end{aligned} \tag{8}$$

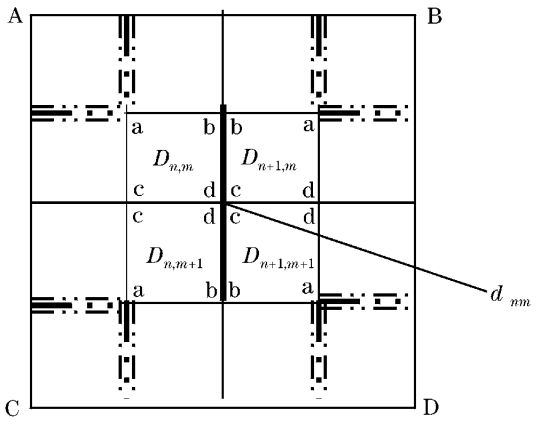

Fig. 1. Illustration of estimating fractal parameters.

where $\overrightarrow{\Delta_1} = \{\Delta_1(i,j)\} = \{P(i+(n-1)*S, j+(m-1)*S) - g_{nm}(i+(n-1)*S, j+(m-1)*S)\}$, $\overrightarrow{\Delta_2} = \{\Delta_2(i,j)\} = \{P(i+n*S, j+(m-1)*S) - g_{nm}(i+n*S, j+(m-1)*S)\}$, $\overrightarrow{\Delta_3} = \{\Delta_3(i,j)\} = \{P(i+(n-1)*S, j+m*S) - g_{nm}(i+(n-1)*S, j+m*S)\}$, $\overrightarrow{\Delta_4} = \{\Delta_4(i,j)\} = \{P(i+n*S, j+m*S) - g_{nm}(i+n*S, j+m*S)\}$, $\overrightarrow{B_1} = \{B_1(i,j)\} = \{P(i*S, j*S) - h(i*S, j*S)\}$, $\overrightarrow{B_2} = \{B_2(i,j)\} = \{B_1(S-i,j)\}$, $\overrightarrow{B_3} = \{B_3(i,j)\} = \{B_1(S-i,j)\}$, $\overrightarrow{B_4} = \{B_4(i,j)\} = \{B_1(S-i,S-j)\}$, $\overrightarrow{I_1} = \{I_1(i,j)\} = \{\Phi_{00}(i,j)\}$, $\overrightarrow{I_2} = \{I_2(i,j)\} = \{\Phi_{00}(S-i,j)\}$, $\overrightarrow{I_3} = \{I_3(i,j)\} = \{\Phi_{00}(i,S-j)\}$, $\overrightarrow{I_4} = \{I_4(i,j)\} = \{\Phi_{00}(S-i,S-j)\}$.

Let $\overrightarrow{a\_1} = \overrightarrow{\Delta_1} - (d_{n-1,m-1}\overrightarrow{I_1} + d_{n-1,m}\overrightarrow{I_2} + d_{n,m-1}\overrightarrow{I_3}) \otimes \overrightarrow{B_1}$, $\overrightarrow{a\_2} = \overrightarrow{\Delta_2} - (d_{n,m-1}\overrightarrow{I_1} + d_{n+1,m-1}\overrightarrow{I_2} + d_{n+1,m}\overrightarrow{I_4}) \otimes \overrightarrow{B_2}$, $\overrightarrow{a\_3} = \overrightarrow{\Delta_3} - (d_{n-1,m}\overrightarrow{I_1} + d_{n-1,m+1}\overrightarrow{I_3} + d_{n,m+1}\overrightarrow{I_4}) \otimes \overrightarrow{B_3}$, $\overrightarrow{a\_4} = \overrightarrow{\Delta_4} - (d_{n+1,m}\overrightarrow{I_2} + d_{n,m+1}\overrightarrow{I_3} + d_{n+1,m+1}\overrightarrow{I_4}) \otimes \overrightarrow{B_4}$, $\overrightarrow{b\_1} = \overrightarrow{I_4} \otimes \overrightarrow{B_1}$, $\overrightarrow{b\_2} = \overrightarrow{I_3} \otimes \overrightarrow{B_2}$, $\overrightarrow{b\_3} = \overrightarrow{I_2} \otimes \overrightarrow{B_3}$, and $\overrightarrow{b\_4} = \overrightarrow{I_1} \otimes \overrightarrow{B_4}$, the value of d_{nm} to minimize E_{nm} is

$$d_{nm} = \frac{\overrightarrow{a\_1} \oplus \overrightarrow{b\_1} + \overrightarrow{a\_2} \oplus \overrightarrow{b\_2} + \overrightarrow{a\_3} \oplus \overrightarrow{b\_3} + \overrightarrow{a\_4} \oplus \overrightarrow{b\_4}}{\overrightarrow{b\_1} \oplus \overrightarrow{b\_1} + \overrightarrow{b\_2} \oplus \overrightarrow{b\_2} + \overrightarrow{b\_3} \oplus \overrightarrow{b\_3} + \overrightarrow{b\_4} \oplus \overrightarrow{b\_4}}. \quad (9)$$

Now LIA can solve the non-separation problem when Collage Theorem is applied to find the appropriate fractal parameters, IFS-based 3D fractal interpolation can solve the inverse problem of elevation data compression. The experiments of DEM file from USGS are shown in Figs. 2 and 3. Figure 2(a) is 1:250,000-scale DEM for Idaho(named kalispell-e) and Fig. 3(b) is another 1:250,000-scale DEM for Idaho (named hailey-e). Efficiency of such algorithm can be measured with the common compression ratio and peak signal-to-noise ratio (PSNR) for the reconstructed DEM.

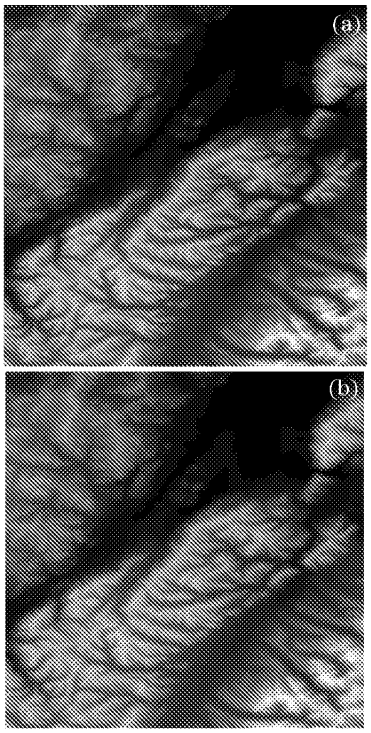

Fig. 2. (a) Original DEM and (b) reconstructed DEM with sample rate 1:8.

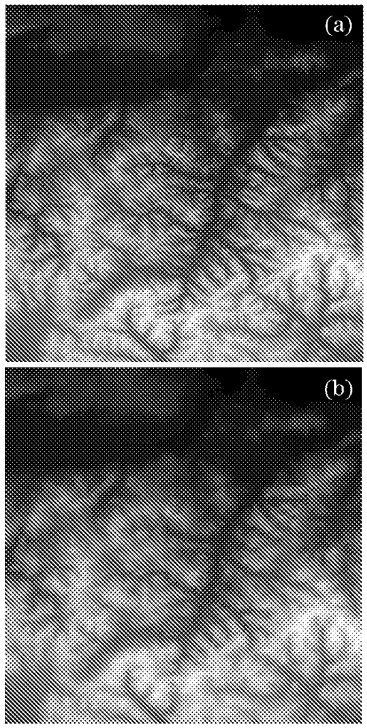

Fig. 3. (a) Original DEM and (b) reconstructed DEM with sample rate 1:16.

The full process of the algorithm is as follows.

1) The DEM file is regularly sampled to gain the pairs of interpolation points set $\{(x_n, y_m, P_{n,m})$, for $n = 0, 1, \cdots, L$, $m = 0, 1, \cdots, L\}$.

2) LIA is used to find the right fractal parameters $\{d^*_{nm}$ for $n = 0, 1, \cdots, L$, $m = 0, 1, \cdots, L\}$.

a) The iteration starts from any initial values $\{d_{nm}\}$ and only changes a single parameter d_{nm} at a time.

b) d^*_{nm} can be obtained from Eqs. (8) and (9) to reach the minimum E_{nm} (the superscript denotes the new value after calculation).

c) Accordingly, the next value of $\{d_{nm}\}$ such as $d^*_{n+1,m}$ can be achieved by replacing d_{nm} with d^*_{nm}. In turn, the other values can be obtained in similar process to reach new fractal values $\{d^*_{nm}\}$, which is gradually converge on final optimization.

3) The pairs of interpolation points set $\{(x_n, y_m, P_{n,m})$, where $n = 0, 1, \cdots, L$, $m = 0, 1, \cdots, L\}$ and fractal parameters $\{d^*_{nm}$, $n = 0, 1, \cdots, L$, $m = 0, 1, \cdots, L\}$, which yield the minimum E for Eq. (6), are stored as the compressed data. The compression rate can be achieved.

4) The reconstructed terrain is the attractor of the above stored IFS using 3D fractal interpolation described in Eqs. (1)–(3). $d(x,y)$ in Eq. (3) can now be expressed by fractal parameters $\{d^*_{nm}$, $n = 0, 1, \cdots, L$, $m = 0, 1, \cdots, L\}$. A multi-resolution recursion is adopted in the process of reconstruction to replace the common time-consuming fractal iteration[8]. The PSNR is calculated for describing the difference between original DEM and reconstructed DEM.

The repeated process of convergence in LIA enjoys advantages over the common block-based fractal compression[9]. The DEM in Fig. 2(a) is sampled by 1:8

and encoded by using the ILA of 3D fractal interpolation. The compression rate is 31.75, while the PSNR is 40.05 dB for the reconstructed DEM in Fig. 2(b). The DEM in Fig. 3(a) is sampled by 1:16 and encoded by using the LIA of 3D fractal interpolation. The compression rate is 126.02, while the PSNR is 30.63 dB for the reconstructed DEM in Fig. 3(b). The results show that the compression rate and PSNR are highly effected by sample rate.

Although the computation of the fractal coding based on 3D fractal interpolation is complicated, Eqs. (8) and (9) can actually be calculated within a short time without the time-consuming searching and matching process for the block fractal coding[9]. Also, LIA enhances the coding process in which each step of iteration is computed locally and quickly. When LIA is considered as a kind of zero-search fractal coding method, it is similar to bath fractal transformation[10] but differs in that the LIA enjoys an explicit geometric meaning described by Eqs. (1)−(3) and higher reconstruction quality. The coding of DEM in Fig. 2(a) has been completed within 3.5 s with the PC computer (CPU PIII550, RAM 128M).

The experimental results prove the efficiency and practicality of using IFS-based 3D fractal interpolation for elevation data compression. Furthermore, the algorithm can also be applied for many fields such as terrain-based mapping and remote-sensing.

S. Wu's e-mail address is wusy@sjtu.edu.cn.

References

1. B. B. Mandelbrot, *The Fractal Geometry of Nature* (Freeman, New York, 1982) chap. 6.
2. Q. H. Lou and F. Huang, Chin. Opt. Lett. **1,** 231 (2003).
3. H. Honda, M. Haseyama, and H. Kitajima, in *Proceedings of IEEE International Conference on Image Processing* 657 (1999).
4. J. R. Price and M. H. Hayes III, IEEE Signal Processing Lett. **5,** 9 (1998).
5. C. M. Wittenbrink, in *Proceeding of the IEEE Visualization Conference* 77 (1995).
6. H. P. Xie, H. Q. Sun, Y. Ju, and Z. G. Feng, International J. Solids and Structures **38,** 32 (2001).
7. G. Kim and A. P. Barros, Remote Sensing of Environment **83,** 2 (2002).
8. S. Y. Wu and Y. H. Zhou, J. Shanghai Jiaotong University **38,** 4 (2004).
9. Y. Fisher, *Fractal Image Compression Theory and Application* (Springer-Verlag, New York, 1995) chap. 3.
10. D. M. Monro and S. J. Woolley, in *Proceeding of IEEE Conference on Acoustics, Speech and Signal Processing* 557 (1994).