

文章编号:1001-9081(2006)11-2701-03

基于函数迭代系统的 3-D 分形插值算法

王 梦, 金文标

(重庆邮电大学 计算机图形图像研究所, 重庆 400065)

(cqwmeng@163.com; jinwb@cqupt.edu.cn)

摘 要:提出了一种新的分形插值算法,通过矩形剖分上的采样数据点构建分形插值曲面。该算法保证分形插值时的边界连续性,而且对于初始数据集没有任何对称性限制。所构建的分形插值曲面整体上保持原始数据的主要特征,局部上具备自相似的特点。实验结果表明算法的有效性和低时间复杂度,有利于分形插值的实际应用。

关键词:分形插值;迭代函数系统;矩形剖分;对称性

中图分类号:TP317.4 **文献标识码:**A

A 3-D fractal interpolation algorithm based on iterated function system

WANG Meng, JIN Wen-biao

(Institute of Computer Graphics and Image Processing, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: A new method was proposed for constructing fractal interpolation surfaces through points sampled on rectangular lattices. The proposed algorithm guaranteed the boundary continuity and canceled symmetry restriction on the initial data set. The constructed surface inherited the main features from the original data set as a whole and kept the self-similar trait of fractal in part. Experiment results show that the method is useful for three-dimensional (3D) fractal interpolation.

Key words: fractal interpolation; iterated function system(IFS); rectangular lattice; symmetry

0 引言

分形插值是美国数学家 M. F. Bamsley^[1] 于 1986 年首次提出来的,他给出了拟合数据的一种新思想,能够较好地刻画自然界中存在的大量不光滑的曲线和曲面,并且广泛应用于各种数据可视化、模型重建和仿真等问题。M. F. Bamsley^[1] 利用了一组仿射变换产生的迭代函数系统(Iterated Function System, IFS)生成分形插值函数(Fractal Interpolation Function, FIF),该函数对平面上的一组节点进行插值。此方法已被用于生成各种二维分形图形^[2,3],但在三维的情形下,由于计算复杂、子域边界连续性等原因,其应用还相当有限。

目前有两种方式可将平面 FIF 扩展到三维空间。P. R. Massopust^[4] 等在 R^3 上引入迭代函数系统来构造分形曲面。他研究了一类基于三角剖分的三维分形插值算法,但该方法与另外一种基于矩形剖分^[5] 的方法相比需要更多的压缩变换,使用也不太方便。文献[5]讨论了基于矩形剖分的三维分形插值理论,但该方法需要对初始数据集的边界进行对称性限制,在实际应用中有相当的局限性;文献[6]在文献[5]的基础上,提出通过改变邻近子域的映射顺序来达到边界连续的目的,但其映射过程复杂,计算量也相对较大。

本文提出的算法可以处理任意的初始数据集,仅需要一种映射方式即可满足边界连续性条件。

1 分形插值函数

对于平面上 FIF,首先给出一个初始的数据集:

$$\{(x_i, y_i) \in D \times R: i \in [0, 1, \dots, N]\} \quad (1)$$

其中 x_i 严格递增,闭区间 $D = [x_0, x_N] \subset R$,希望构造一

个插值函数 $f: D \rightarrow R$,使得:

$$f(x_i) = y_i, i \in [0, 1, \dots, N] \quad (2)$$

对于每一个子区间定义如下仿射变换:

$$w_i \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_i & 0 \\ c_i & d_i \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix}, i \in [0, 1, \dots, N] \quad (3)$$

为了插值给定的原始数据点,需要满足以下端点条件:

$$\begin{cases} w_n \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} x_{i-1} \\ y_{i-1} \end{pmatrix} \\ w_n \begin{pmatrix} x_N \\ y_N \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \end{pmatrix} \end{cases}, i \in [0, 1, \dots, N] \quad (4)$$

每一个映射 w_n 将原来在区间 D 上的整个图像压缩到子区间 $D_n = [x_{n-1}, x_n]$ 上,其中水平压缩比由(3)式中的 a_i 决定,而垂直压缩比由 d_i 决定。为了保证收敛,选择 d_i 为自由参数,且 $|d_i| < 1$,由式(1)~(4)计算可得:

$$\begin{cases} a_i = \frac{x_i - x_{i-1}}{x_N - x_0} \\ c_i = \frac{y_i - y_{i-1}}{x_N - x_0} - d_i \frac{y_N - y_0}{x_N - x_0} \\ e_i = \frac{x_N x_{i-1} - x_0 x_i}{x_N - x_0} \\ f_i = \frac{x_N y_{i-1} - x_0 y_i}{x_N - x_0} - d_i \frac{x_N y_0 - x_0 y_N}{x_N - x_0} \end{cases} \quad (5)$$

2 矩形剖分的函数迭代系统

借鉴文献[5]中将区间上一元分形插值算法平行推广到矩形域上二元分形插值的做法,提出了基于矩形剖分的三维

收稿日期:2006-05-18;修订日期:2006-08-22 基金项目:重庆市自然科学基金资助项目(CSTC2004BB7166)

作者简介:王梦(1982-),男,四川遂宁人,硕士研究生,主要研究方向:分形插值理论、数字图像处理;金文标(1964-),男,浙江金华人,副教授,博士,主要研究方向:计算机图形学、计算几何。

分形插值格式,主要定义如下:

在矩形域 $D = I \times J = \{(x, y), p_1 \leq x \leq p_2, q_1 \leq y \leq q_2\}$ 上给出个矩形剖分:

$$p_1 = x_0 < x_1 < \cdots < x_N = p_2$$

$$q_1 = y_0 < y_1 < \cdots < y_N = q_2$$

在此矩形剖分上定义插值数据集: $\{(x_n, y_m, z_{nm}), n = 0, 1, \cdots, N; m = 0, 1, \cdots, M\}$ 。

令 $I_n = [x_{n-1}, x_n], J_m = [y_{m-1}, y_m]$, 子域 $D_{n,m} = I_n \times J_m, n = 1, 2, \cdots, N, m = 1, 2, \cdots, M$ 。

由式(6)定义 IFS 仿射变换 $\{w_{nm}\}$:

$$w_{n,m} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} L_n(x) \\ L_m(y) \\ F_{nm}(x, y, z) \end{pmatrix} \quad (6)$$

其中 $L_n: I \rightarrow I_n, L_m: J \rightarrow J_m$ 是同胚压缩变换,

$$\begin{cases} L_n(x) = x_{n-1} + \frac{x_n - x_{n-1}}{x_N - x_0}(x - x_0) \\ L_m(y) = y_{m-1} + \frac{y_m - y_{m-1}}{y_M - y_0}(y - y_0) \end{cases} \quad (7)$$

$F_{n,m}: R^3 \rightarrow R$ 是定义在矩形域 $D = I \times J$ 上的连续函数:

$$F_{n,m}(x, y, z) = a_{n,m}x + b_{n,m}y + c_{n,m}xy + d_{n,m}z + e_{n,m} \quad (8)$$

同时,为了满足插值的条件,需要该变换满足式(9):

$$\begin{cases} F_{n,m}(x_0, y_0, z_{00}) = z_{n-1, m-1} \\ F_{n,m}(x_N, y_0, z_{N0}) = z_{n, m-1} \\ F_{n,m}(x_0, y_M, z_{0M}) = z_{n-1, m} \\ F_{n,m}(x_N, y_M, z_{NM}) = z_{n, m} \end{cases} \quad (9)$$

选择 $d_{n,m}$ 为自由变量,根据以上 4 个条件,计算可得:

$$\begin{cases} c_{n,m} = [z_{n-1, m-1} - z_{n-1, m} - z_{n, m-1} - \\ d_{n,m}(z_{0,0} - z_{0,M} - z_{N,0} + z_{N,M})] / \\ (x_0y_0 - x_Ny_0 - x_0y_M + x_Ny_M) \\ a_{n,m} = [z_{n-1, m-1} - z_{n, m-1} - d_{n,m}(z_{0,0} - z_{N,0}) - \\ c_{n,m}(x_0y_0 - x_Ny_0)] / (x_0 - x_N) \\ b_{n,m} = [z_{n-1, m-1} - z_{n-1, m} - d_{n,m}(z_{0,0} - z_{0,M}) - \\ c_{n,m}(x_0y_0 - x_0y_M)] / (y_0 - y_M) \\ e_{n,m} = z_{n,m} - a_{n,m}x_N - b_{n,m}y_M - \\ c_{n,m}x_Ny_M - d_{n,m}z_{N,M} \end{cases} \quad (10)$$

分形插值曲面即是上述 IFS 的吸引子^[5]。

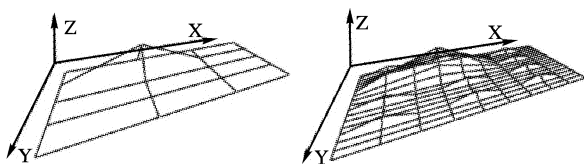


图 1 对有边界对称性条件限制的数据集分形插值示意图

下面给出一个例子来说明使用上述算法进行 3-D 分形插值。初始的数据集是在空间中的 20 个点,其坐标值如表 1 所示。该矩形区域边界数据点的 Z 值都为 0,满足将在下节详细说明的对称性条件(11)。图 1(a)是将未经过插值处理的初

表 1 分形插值初始数据集

Y	X			
	0	80	160	240
0	0	0	0	0
20	0	30	0	0
40	0	0	0	0
60	0	0	0	0
80	0	0	0	0

始点、邻近点用直线连接。图 1(b)是使用上述算法对初始点进行一次分形插值迭代后的结果,其中参数 $d = 0.2$ 。显然,图 1(b)所示的曲面具有分形自相似的特点。

3 改进的基于矩形剖分 IFS 算法

构建 FIS 的最大困难在于保证边界的连续性。因为邻近的子域由不同的映射得到,其公共边界上相同位置的插值点就会对应不同的 Z 值。如图 2,以子域 D_n 和 D_m 公共边上的点 X 为例,在压缩变换过程中,如果在子域 D_n 上所生成的点 X 的 Z 值与子域 D_m 在该点的 Z 值不同,就会造成边界的不连续。要使边界连续,必须保证相邻子域公共边上的插值点有相同 Z 值。

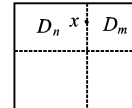


图 2 矩形剖分分形插值的矩形域和子域的边界



图 3 对初始矩形数据域进行特定扩展使其具有边界对称性

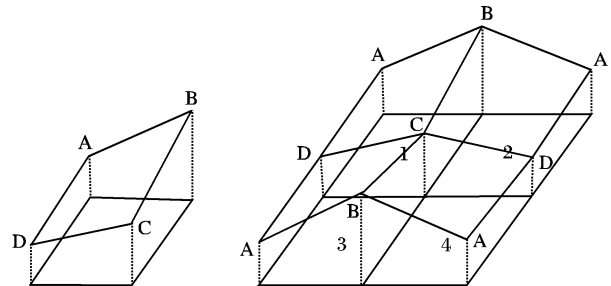


图 4 图 3 的三维立体示意图

为了利用第 2 节中所述算法得到一个连续的插值曲面,初始数据集需要满足以下条件:

$$\begin{cases} z_{0,m} = z_{N,m}, m = 0, 1, \cdots, M \\ z_{n,0} = z_{n,M}, n = 0, 1, \cdots, N \end{cases} \quad (11)$$

以图 3(a)中所示的矩形数据域为例,若满足条件(11),即是要求其上边界 AB 和下边界 DC 对称,同时要求左边界 AD 与右边界 BC 对称。这样才能使相邻子域在压缩变换过程中,在其公共边上相同位置的插值点生成相同的 Z 值,从而得到连续的插值曲面。表 1 中的数据是满足这个条件的,故可使用第 2 节所述算法。但这样的条件太苛刻,不适合实际应用。在大多数情况下,采样得到的边界数据点的 Z 值是任意的。为此,本文提出一种改进的基于矩形剖分 IFS 算法,以处理任意边界的矩形剖分的数据集。

图 3(a)给出了一个任意高度值的矩形数据域,图 4(a)是该矩形域上的 3-D 视图(增加了高度值),其上边界 AB 和下边界 DC,左边界 AD 和右边界 BC 没有任何对称性限制。本算法基本的步骤如下:

1) 扩展初始的矩形数据域为原来的 4 倍扩展的方式,如图 3 所示。该扩展不是简单的放大原矩形域,而是先作原矩形域的镜像图像,再拼接成面积为原来 4 倍的矩形域。图 3(b)中,4 块矩形域,分别标号为 1、2、3 和 4。第 1 块就是原矩形域,第 2 块是原矩形域的水平镜像,第 3 块是原矩形域的垂直镜像,第 4 块是第 3 块的水平镜像。这样,得到一个更大

的矩形域,该矩形显然满足前述条件(11)。即它的上边界 ABA 与下边界 ABA 对称,同理左边界 ADA 与右边界 ADA 也对称。图 4 是图 3 的立体效果图,它们都反映了如何进行初始域扩展的方式。

2) 对这个扩展后的矩形数据域,使用第 2 节中的算法进行分形插值,得到的插值曲面作为中间曲面。

3) 对于该中间曲面裁剪其 1/4 的区域作为最后的结果曲面,具体裁剪的位置如图 3(b)中标号为 1 的那一块区域。

4 实验

对下面的例子用本文的改进方法来进行分形插值。其初始数据点也有 20 个点,设矩形域 $D = [0, 450] \times [0, 280]$, $N = 3$, $M = 4$, $d_{n,m} = 0.2$ 。其插值点坐标值如表 2 所示: $\{(x_n, y_m, z_{nm}), n = 0, 1, 2, 3; m = 0, 1, \dots, 4\}$ 。

与表 1 的数据相比较,表 2 的数据没有边界对称性限制。

图 5(a)是将表 2 的数据直接绘制在 3D 坐标系中;(b)是扩展后的新矩形域,该矩形域具有边界对称性;(c)是对于扩展后的新矩形域应用第 2 节的算法进行一次分形插值;(d)是裁剪(c)中所示曲面的 1/4 作为最后结果曲面。

Y	X			
	0	150	300	450
0	50	80	60	20
70	61	205	80	35
130	40	140	55	35
210	30	50	40	28
280	42	45	50	40

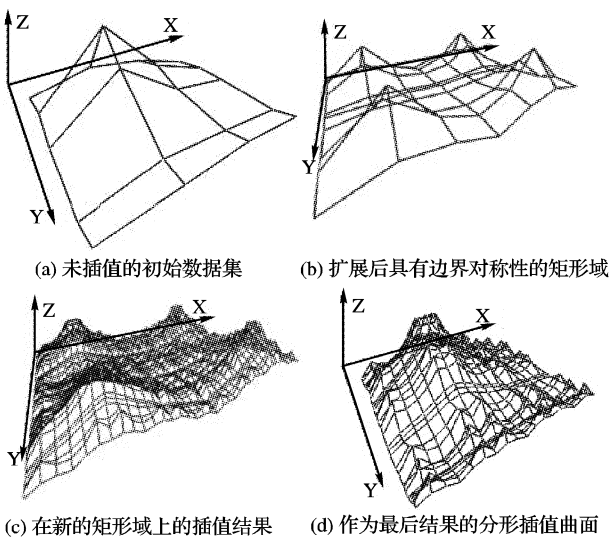


图 5 对无边界条件限制的数据集使用本文算法进行分形插值的过程和结果曲面

5 结语

本文所提出的改进算法,能够对矩形域上的任意数据进行分形插值从而得到连续的分形曲面。算法的关键是扩展初始数据域,最后又裁剪迭代完成的曲面,使得最后的结果曲面满足插值条件并保持了分形的自相似的特性。实验结果表明了算法的有效性,并且由于该算法可以处理任意的矩形剖分数据,使得它具有相当的实用性。但是由算法可以看出,该方法具有一定的空间复杂度,即在整个插值过程中需要原来 4 倍的存储空间来存放扩展的数据点。

参考文献:

- [1] BARNESLEY MF. Fractal functions and interpolation[J]. Constructive Approximation, 1986, 2(2): 303-329.
- [2] PRICE JR, HAYES III MH. Resampling and reconstruction with fractal interpolation functions[J]. IEEE Signal Processing Letters, 1998, 5(9): 228-230.
- [3] WEN J, ZHU X. An improved direct inverse problem solver for fractal interpolation functions with applications to signal compression[A]. Proceedings of Signals, Systems, and Electronics'95[C], 1995: 187-190.
- [4] MASSOPUST PR. Fractal surface[J]. Journal of mathematical analysis and application, 1990, 151(1): 275-290.
- [5] 齐东旭. 分形及其计算机生成[M]. 北京: 科学出版社, 1994. 68-75.
- [6] 吴思源, 周源华. 基于三维 IFS 分形插值的 DEM 重构算法[J]. 上海: 上海交通大学学报, 2004, 4, 38(4): 537-541.
- [7] PRICE JR, HAYES III MH. Fractal interpolation of images and volumes[A]. Proceedings of the Thirty-Second Asilomar Conference on Signals, Systems, and Computers[C], 1998.
- [8] TEBOURBI R, BELHADJ Z, ZRIBI M, et al. 3-D soil reconstruction from binocular stereo disparity[A]. Proceedings of Geosience and Remote Sensing Symposium 2000[C], 2000.
- [9] XIE HP, SUN HQ. The study of bivariate of fractal interpolation function and creation of fractal interpolation surface[J]. Fractal, 1997, 5(4): 625-634.
- [10] WITTENBRINK CM. IFS fractal interpolation for 2D and 3D visualization [A]. Proceedings of IEEE Visualization Conference[C], 1995. 77-84.
- [11] BARNESLEY MF. Fractal image compression[M]. Wellesley massachusetts: AK Peter Ltd, 1993. 173-218.
- [12] QIAN X-Y. Bivariate fractal interpolation functions on rectangular domains[J]. Journal of computational mathematics, 2002, 20(4).
- [13] DUDGEON JE, GOPALAKRISHNAN R. Fractal based modelling of 3D terrain surfaces[A]. Proceedings of Bringing Together Education, Science and Technology'96[C], 1996, 4: 246-252.
- [14] MARAGOS P. Fractal aspects of speech signals, dimension and interpolation[A]. ICASSP[C], 1991. 1. 417-420.

(上接第 2700 页)

- [2] UYSAL M, YARMAN - VURAL FT. A Fast Color Quantization Algorithm Using a Set of One Dimensional Color Intervals[A]. Proceedings of ICIP'98[C], 1998. 191-195.
- [3] HECKBERT P. Color image quantization for frame buffer display[J]. ACM Transactions on Computer Graphics, 1982, 16(3): 297-307.
- [4] SCHEUNDERS P. A comparison of clustering algorithms applied to color image quantization[J]. Pattern Recognition Letters, 1997, 18: 1379-1384.
- [5] BRAQUELAIRE JP, BRUN, L. Comparison and Optimization of Methods of Color Image Quantization[J]. IEEE Transactions on Image Processing, 1997, 6(7): 1048-1052.
- [6] SIRISATHITKUL Y, AUWATANAMONGKOL S, UYYANONVARA B. Color image quantization using distances between adjacent colors along the color axis with highest color variance[J]. Pattern Recognition

Letters, 2004, 25: 1025-1043.

- [7] SHARMA G, TRUSSELL HJ. Digital Color Imaging[J]. IEEE Transactions on Image Processing, 1997, 6(7): 901-932.
- [8] LINDE Y, BUZO A, GRAY R. An algorithm for vector quantizer design[J]. IEEE Transactions on Communications, 1980, 28(1): 84-95.
- [9] 刘春和, 陆哲明, 孙圣和. 等均值等范数最近邻矢量量化码字搜索算法[J]. 电子学报, 2003, 31(10): 1558-1561.
- [10] HUANG CM, BI Q, STILES GS, HARRIS RW. Fast full search equivalent encoding algorithms for image compression using vector quantization[J]. IEEE Transactions on Image Processing, 1992, 1(3): 413-416.
- [11] ZHANG X, SONG ZM, WANG YL, et al. Color Quantization of Digital Images[A]. Proceedings of PCM2005, LNCS 3768[C], 2005. 653-664.