

纹理和 Fractals

蔡立农

(电子技术系)

摘 要

文中着重讨论了三维逼真性图形显示技术中的二个问题: 1. 纹理处理; 2. fractal方法及其在计算机图形学中的应用。纹理处理是在三维明暗图形的基础上, 模拟三维物体表面细节的过程, 纹理处理分为两类: (1) 对光滑表面; (2) 对粗糙表面。fractal方法是描述不规则物体的一种较新的方法, 文中较详细地讨论了fractal方法及其在计算机图形学中的应用, 给出了模拟地形的方法和处理结果。

关键词: 逼真性, 纹理处理, fractals

1 引 言

在计算机图形学中, 显示三维逼真性图形是最活跃、最新的研究领域。当将其用于CAD中, 就会使CAD在国防、科学技术、经济建设和日常生活中, 产生极大的作用。

一般在计算机显示屏上显示三维逼真性图形, 要进行以下几种处理: 1. 物体的几何表示; 2. 几何变换和投影变换; 3. 消隐面; 4. 明暗处理; 5. 阴影处理; 6. 纹理处理; 等等。

本文着重对三维逼真性图形显示中的二个问题进行研究: 1. 纹理处理; 2. fractal方法及在计算机图形学中的应用。

2 显示三维逼真性图形的技术

2.1 三维物体的表示

在计算机图形学中, 物体的几何表示是进行三维图形逼真显示的基础。三维物体在计算机内常用的表示方法有: 线框模型、表面模型、实体模型和彩色分域模型。本文采用表面模型来表示三维物体, 主要有: 球、圆柱、圆锥和Bezier曲面。

2.2 明暗处理

- 1) 采用Phong的光照模型^[1]。
- 2) 光线跟踪法。

用光线跟踪法对物体进行明暗处理的基本思想是: 用视点与屏面上一点的射线去跟踪物体, 求出射线与物体表面的诸可能的交点, 将这些交点按深度排序, 看谁离视点近。

找出这一可见点后, 计算其明暗值并加以显示。如果考虑透射、折射等, 还需进一步进行光线跟踪。作者在显示球、圆柱、圆锥明暗图形时, 因为求交只需解一元二次方程, 故采用了此方法。

3) 物体可见优先级排序法。

用光线跟踪法显示的三维图形非常逼真。但它有个致命弱点: 计算时间长; 如果物体表面复杂, 还需求解高阶方程, 比如Bezier曲面。本文对它进行明暗处理时, 采用了物体可见优先级排序法。该方法的基本思想是: 先将物体表面多次再分, 直至分为近似一点; 求出该点的明暗值; 然后将这些所有物体的可见点, 根据深度值优先级排序, 最后显示那些离视点近的可见点。

3 纹理处理

有了三维明暗图形以后, 为了使图形更加逼真, 还要对物体表面的细节进行模拟, 亦即纹理处理。纹理处理分为两类: 一种是对光滑物体的表面, 另一种是对粗糙物体的表面。

3.1 光滑表面的纹理处理

光滑表面的纹理处理, 实际上就是将纹理样本映射到光滑表面上, 纹理色彩值与表面的色彩值进行加权平均, 处理后的物体表面仍呈现光滑。

纹理映射过程实质上就是物空间到纹理空间, 再到象空间的变换过程。如果纹理样本在单位纹理空间上用 x_i, y_i 表示, 且 $0 \leq x_i \leq 1, 0 \leq y_i \leq 1$, 物体在物空间上用 θ, φ 来表示, 那么纹理映射就是确定两空间的变换函数表达式。设 $\theta = f(x_i, y_i), \varphi = g(x_i, y_i)$, 或相反 $x_i = r(\theta, \varphi), y_i = s(\theta, \varphi)$ 。

一般取线性映射, 否则图形会产生失真。设线性映射函数为

$$\theta = Ax_i + B, \quad \varphi = Cy_i + D,$$

其中 A, B, C, D 为待定常数。

只要给出物体的数学表示式, 就可确定 A, B, C, D 值, 从而确定纹理空间到物空间的变换表达式, 再求出物空间至纹理空间的变换表达式。

如圆柱体, 参数表示式为

$$\begin{cases} x = x_0 + R \cos \theta \\ y = y_0 + h \\ z = z_0 + R \sin \theta \end{cases} \quad \text{或相反} \quad \begin{cases} \theta = \text{tg}^{-1} \left(\frac{z - z_0}{x - x_0} \right) \\ h = y - y_0 \end{cases}$$

如果只对圆柱体的一部分进行纹理映射, 那么取 $-\pi \leq \theta \leq 0$ 和 $0 \leq h \leq h_{\max}$ 。

因为变换过程中, 四个顶点对应, 那么当 $(x_i, y_i) = (0, 0)$ 时, $(\theta, h) = (-\pi, h_{\max})$; $(x_i, y_i) = (0, 1)$ 时, $(\theta, h) = (-\pi, 0)$; $(x_i, y_i) = (1, 0)$ 时, $(\theta, h) = (0, h_{\max})$; $(x_i, y_i) = (1, 1)$ 时, $(\theta, h) = (0, 0)$ 。根据上面条件, 可得变换式为:

$$\begin{cases} \theta = \pi x_i - \pi \\ h = h_{\max} - y_i h_{\max} \end{cases}$$

再得

$$\begin{cases} x_i = (\theta + \pi) / \pi \\ y_i = (h_{\max} - h) / h_{\max} \end{cases}$$

本文将红白网格状纹理, 映射到了圆柱体上。

3.2 粗糙表面的纹理处理

即使将粗糙纹理的样本映射到物体的表面, 显示出的图形也不是很令人满意。它看起来好象是把粗糙纹理样本画到了物体表面似的。其原因是, 具有粗糙表面的物体, 它的表面法线矢量是不规则的。

解决的办法是: 对表面法线矢量进行扰动。设物体表面任一点 $\vec{P} = (x, y, z)$, 且 $X = x(u, v)$, $Y = y(u, v)$, $Z = z(u, v)$, \vec{P} 在 u, v 方向上的偏导分别为 \vec{P}_u 和 \vec{P}_v 。

设表面法矢 $\vec{N} = \vec{P}_u \times \vec{P}_v$ 。

用扰动函数 $F(u, v)$ 对原表面法线进行扰动, 那么物体表面经扰动后的一点为

$$\vec{P}^* = \vec{P} + F \cdot \vec{N} / |\vec{N}|$$

偏导矢量分别为

$$\vec{P}_u^* = \vec{P}_u + F_u \cdot \vec{N} / |\vec{N}| + F \cdot (\vec{N} / |\vec{N}|)_u$$

$$\vec{P}_v^* = \vec{P}_v + F_v \cdot \vec{N} / |\vec{N}| + F \cdot (\vec{N} / |\vec{N}|)_v$$

因为 F 值很小, 可以略去后一项。

$$\vec{P}_u^* \doteq \vec{P}_u + F_u \cdot \vec{N} / |\vec{N}|, \quad \vec{P}_v^* \doteq \vec{P}_v + F_v \cdot \vec{N} / |\vec{N}|$$

扰动后的表面法线矢量

$$\vec{N}^* = \vec{N} + \vec{D}$$

其中

$$\vec{D} = \frac{[F_u(\vec{N} \times \vec{P}_v) - F_v(\vec{N} \times \vec{P}_u)]}{|\vec{N}|}$$

对于 $F(u, v)$ 的选择, 可以是确定的数学表示式, 也可以是随机函数, 或查找表。采用确定的数学表示式, 如考虑球体, 其参数表示式为:

$$x - x_0 = R \cos \theta \cdot \cos \varphi$$

$$y - y_0 = R \cdot \sin \theta \cdot \cos \varphi$$

$$z - z_0 = R \cdot \sin \varphi$$

那么

$$\theta = \arctg[(y - y_0) / (x - x_0)]$$

$$\varphi = \arctg[\pm(z - z_0) / \sqrt{(x - x_0)^2 + (y - y_0)^2}]$$

表面法线

$$\vec{N} = \frac{\partial \vec{P}}{\partial \theta} \times \frac{\partial \vec{P}}{\partial \varphi} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ -R \cdot \sin \theta \cdot \cos \varphi & R \cos \theta \cdot \cos \varphi & 0 \\ -R \cdot \cos \theta \cdot \sin \varphi & -R \cdot \sin \theta \cdot \sin \varphi & R \cos \theta \end{vmatrix}$$

$$= R^2 \cdot \cos \theta \cdot \cos^2 \varphi \cdot \vec{i} + R^2 \cdot \sin \theta \cdot \cos^2 \varphi \cdot \vec{j} + R^2 \sin \varphi \cos \theta \cdot \vec{k}$$

$$|\vec{N}| = R^2 \cdot \cos \varphi$$

取

$$F(\theta, \varphi) = K_1 \cdot \sin K_2 \theta \cdot \sin K_2 \varphi$$

那么

$$\vec{D} = [F_\theta(\vec{N} \times \vec{P}_\varphi) - F_\varphi(\vec{N} \times \vec{P}_\theta)] / |\vec{N}|$$

$$= K_1 K_2 R (\sin \theta \cdot \cos K_2 \theta \cdot \sin K_2 \varphi$$

$$\begin{aligned}
 & + \sin\varphi \cdot \cos\varphi \cdot \cos\theta \cdot \sin K_2\theta \cdot \cos K_2\theta) \cdot \vec{i} \\
 & + K_1 K_2 R (-\cos\theta \cdot \cos K_2\theta \cdot \sin K_2\varphi \\
 & + \sin\varphi \cdot \cos\varphi \cdot \sin\theta \cdot \sin K_2\theta \cdot \cos K_2\varphi) \cdot \vec{j} \\
 & + K_1 K_2 R (-\cos^2\varphi \cdot \sin K_2\theta \cdot \cos K_2\varphi) \cdot \vec{k} \\
 & \vec{N}^* = \vec{N} + \vec{D}
 \end{aligned}$$

由此显示的图形, 孔形纹理较明显。

4 fractal方法及其应用

50年代初, 美国的B.B.Mandelbrot发现了一种几何结构, 可以用来描述自然界中不规则的物体, Mandelbrot用fractal来表示。fractal来自拉丁文fractus, 它与英语中fracture和fragment具有相同的词根, 有“不规则”和“碎片”的意思。研究fractals (fractal集合) 的几何学, 又叫fractal几何, 它涉及了几何拓扑学、集合论和随机过程等。fractal方法不仅应用于计算机图形学中, 而且应用于近代物理、通讯、模式和语音识别、市场商品价格浮动上等方面。

4.1 fractals

定义: fractals定义为一个集合, 该集合的fractal维数 (即Hausdorff—Besicovich维) 应严格大于拓扑维数。

凡是满足上述定义的任何集合, 都是fractal模型。这里涉及fractal维数。

因为fractal模型隐含了细节的无穷递归和图形不断放大时会显露出更多的细节, 那么fractal具有两个重要的性质:

- 1) 比例不变性。无论用什么样的比例去观察模型, 仍象原模型。
- 2) 自相似性。模型的任何一部分都相似于整个模型。

Mandelbrot的fractal维数定义: 根据fractal模型的自相似性, 不妨假设, 把一直线段分成 N 段, 那么划分的每段长度与原直线段长度的比率之和等于1, 即

$$\sum_{s=1}^N r_s = 1$$

此时是直线, 维数取1。如果考虑的是fractal模型, 则应给上面的每一比率加上幂指数 D (D 为实数, 直线时 $D=1$), 即 $\sum_{s=1}^N (r_s)^D = 1$ 。

因为要考虑fractal模型的自相似性, 所以划分的每段均相同, $N \cdot (r_s)^D = 1$

所以fractal维数: $D = \log N / \log\left(\frac{1}{r_s}\right)$ 。

当 $0 < D < 1$, 是fractal点集; 当 $1 < D < 2$, 是fractal曲线; 当 $2 < D < 3$, 是fractal表面。 D 的大小直接反映了fractal点集、曲线和表面的粗糙程度。

采用对代数函数的迭代方法而产生的fractal模型, 能在屏上显示美丽的图案, 它适用于装潢设计和计算机艺术等领域。而fBm调制的fractal模型, 则可产生粗糙表面的纹理和显示海岸线和地形的逼真图形。

4.2 fBm

fBm(fractional Brownian motion)是一种随机函数。函数定义如下:

u 为实参数, 且 $-\infty < u < +\infty$, w 是采样空间 W 中的随机函数值的集合。

$B(u, w)$, 原 Brownian motion, 是独立的高斯随机实函数, 其增量 $B(u_2, w) - B(u_1, w)$ 是零均值, 方差为 $|u_2 - u_1|$ 的高斯随机函数。

设有 h , 且 $0 < h < 1$, b_0 为任意实数, 随机函数 $B_h(u, w)$, 叫降低的 fractional Brownian motion。 $B_h(u, w)$ 定义如下:

$$B_h(0, w) = b_0$$

$$B_h(u, w) - B_h(0, w) = \frac{1}{\Gamma\left(h + \frac{1}{2}\right)}$$

$$\left\{ \int_{-\infty}^0 [(u-s)^{h-\frac{1}{2}} - (-s)^{h-\frac{1}{2}}] dB(s, w) + \int_0^u (u-s)^{h-\frac{1}{2}} dB(s, w) \right\}$$

$B_h(u, w)$ 是由 $(u-s)^{h-\frac{1}{2}}$ 加权的 $B(u, w)$ 平均值, 并且它是可变的。当 $h = \frac{1}{2}$ 时,

$B_h(u, w)$ 就是原 Brownian motion $B(u, w)$ 。这样可得一系列依赖于 u 值的随机函数, fBm 的增量是自相似的, 它的均值 $B_h(u + \Delta u, w) - B_h(u, w)$ 和 $t^{-h} \cdot [B_h(u + t\Delta u, w) - B_h(u, w)]$ 具有相同的联合分布函数。 h 叫自相似参数。

递归中分算法中的 fBm。

下列式子列出了 $B_h(u, w)$ 的条件期望值:

设 $B_h(u, w) = 0$

$$\frac{E[B_h(Ku, w) | B_h(u, w)]}{B_h(u, w)} = \frac{1}{2} [K^{2h} + 1 - |K - 1|^{2h}]$$

当 $K = \frac{1}{2}$ 时,

$$\frac{E\left[B_h\left(\frac{u}{2}, w\right) \middle| B_h(u, w)\right]}{B_h(u, w)} = \frac{1}{2}$$

当 $B_h(0, w) = 0$ 时, $B_h\left(\frac{u}{2}, w\right)$ 的条件期望值为 $\frac{1}{2} B_h(u, w)$, 与 h 无关。自相似性

体现在方差上。故递归中分算法中可产生相应的 fBm。

如果 $N(\mu, \sigma)$ 表示均值为 μ 、方差为 σ 的高斯变量, 假设 fBm 的一维随机函数为 $f(u)$, 则产生 $f(u)$ 的方法如下:

$$1) f(u_{\min}) = N(0, \sigma) = f_1, f(u_{\max}) = N(0, \sigma) = f_2$$

$$2) f\left(\frac{u_{\min} + u_{\max}}{2}\right) = N\left(\frac{f_1 + f_2}{2}, 2^{-h} \cdot \sigma\right)$$

再对 $\left(u_{\min}, \frac{u_{\min} + u_{\max}}{2}\right)$ 和 $\left(\frac{u_{\min} + u_{\max}}{2}, u_{\max}\right)$ 中分下去, 直到满足要求为止。每次

中分中, 可得到相应的fBm. 同理可得到二维 fBm.

4.3. 产生逼真的地形

一般在描述三维地形时, 采用体素模型表示法, 它可用八叉树编码后, 进行处理和显示.

但是这样数据量很大. 可以把描述地形的体素间隔划分得大些, 对每一个体素间隔用fBm调制的方法产生地形的逼真表面.

产生逼真的地形表面, 实际上就是在三维空间上对多边形多次中点再分, 在中点的法线方向用fBm进行一段位移. 这样分下去, 直到满意为止.

如果对平面三角形中点再分, 如图1所示. 三角形ABC, 每边中点分别随机位移移至T、S、R点, 分成了四个小三角形ART、TSC、RSB和RTS, 再对这四个三角形中点再分, 直到达到要求.

如果两个相邻三角形的法线方向不一致, 随机位移后就会出现“裂口”现象. 解决的一种方法, 是对两三角形公共边的两条法线取加权平均. 实际模拟地形时, 法线方向与重力方向平行, 这样每个三角形的法线方向是一致的. 随机位移随着中点再分三角形的变小也减少.

如果对 $y_i (i=1, 2, 3)$ 进行随机位移, 那么位移 $d_i (i=1, 2, 3)$ 就为均值 $=\frac{y_j+y_k}{2}$ ($j, k=1, 2, 3$)、方差 $=\left(\frac{l_i}{2}\right)^{2h}$ 的高斯随机变量, l_i 为三角形 i 边的边长.

这里采用了物体可见优先级排序法, 对模拟的地形进行明暗处理.

5 处理结果

本工作是在国防科大电子技术系研制的 X500 显示系统上完成的. 显示处理机是 Intel 86/330A 微机系统, 三维图形模式下, 显示器分辨率为 $512 \times 512 \times 24$. 程序用 PASCAL 语言编制, 处理结果有照片.

本文由于涉及面较广 (因为篇幅所限, 有些未写入本文), 研究只是初步的, 还存在一些不足, 它将在研究和实践中不断提高, 恳请国内外同行和前辈批评指教.

感谢易晓东教授和陈振初副教授两位老师, 所给予我的精心指导和帮助, 也感谢 405 机房工作人员的支持和帮助.

参 考 文 献

- [1] [美] D. F. 罗杰斯著; 梁友栋等译. 计算机图形学的算法基础. 北京: 科学出版社, 1987
- [2] B.B.Mandelbrot. Fractals: Form, Chance, and Dimension. Freeman, San Francisco, 1977
- [3] J.F.Blinn and M.E.Newell. Comm. ACM; 19(10): 542
- [4] A.Fournier, D.Fussell and L.Carpenter. Comm.ACM, 25(6): 371
- [5] A.Norton.Computer Graphics, 16(3):61
- [6] 蔡立农. 中国自动化学会计算机图形学及辅助设计第二届学术年会, 1987; 79

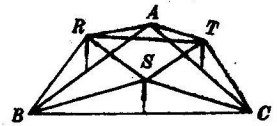


图 1

English Abstracts

Textures and Fractals

Cai linong

Abstract

Two problems in 3-D realistic graphical techniques are discussed in detail in this paper: texturing, and the fractal method and its application to Computer Graphics. Texturing is the process of simulation of the surface details of objects on the basis of 3-D shaded pictures. There are two kinds of texturing. The first is applied to smooth surface while the second is applied to rough one. The fractal method is a new method of describing the irregular object. The writer's opinion on it is presented in the paper. The method of simulating the terrain and the processing results are given as well.

Key words: Realism, Texturing, Fractals.

Design of the Changable Horizontal Frequency Scanner in Monitors

Yu Lifu Li Deqian Qiu Shujia

Abstract

This paper discusses some of the circuit schemes and the calculation methods of the changeable horizontal frequency scanner. By means of the study of some practical examples and experimental implementations, in it are presented the main points of the design methods of horizontal scanners with continually changeable frequency.

Key words: Changable horizontal frequency, Horizontal amplitude, Resolution