

基于硬件加速的纹理映射体绘制

许庆功, 刘庆伟, 张永胜

XU Qing-gong, LIU Qing-wei, ZHANG Yong-sheng

洛阳理工学院 计算机系, 河南 洛阳 471003

Department of Computer Science, Luoyang Institute of Science and Technology, Luoyang, Henan 471003, China

XU Qing-gong, LIU Qing-wei, ZHANG Yong-sheng. Texture mapping volume rendering based on hardware-accelerated. *Computer Engineering and Applications*, 2009, 45(26): 190-192.

Abstract: The rendering process and its importance to computer graphics are introduced. A new method to improve the 3D texture mapping by using multi-textures and dependent-textures is presented and its application in rendering process is studied. It shows that the method has advantage in speed, rebuilding and rendering process and it possesses significance in improving rendering process efficiency.

Key words: volume rendering; dependent texture; texture mapping

摘要:介绍了体绘制的概念及其在计算机绘图中的重要性。提出了利用图形显示卡的相关性纹理和多重纹理映射功能对三维纹理进行有效改进的新算法,研究了该算法在体绘制中的应用,结果表明,它在快速、准确重建和交互性方面具有优势,对于提高体绘制的效率具有重要的意义。

关键词:体绘制;相关纹理;纹理映射

DOI: 10.3778/j.issn.1002-8331.2009.26.057 文章编号: 1002-8331(2009)26-0190-03 文献标识码: A 中图分类号: TP391

1 引言

随着计算机图形硬件性能的飞速提升,利用高性能的图形硬件进行交互,高质量的三维体数据可视化成为可能。纹理映射体绘制算法具有应用广泛、速度快、交互性好的优点,并逐渐成为了研究的热点。三维体数据可视化算法主要有两类:第一类算法是由三维空间数据场构造出中间几何图元,然后再由传统的计算机图形学技术实现画面绘制。虽然这种方法不能反映整个原始数据场的全貌和细节,但是可以产生比较清晰的等值面图像,这是一种常见的可视化算法。其中 Marching Cubes^[1]算法是一种具有代表性的算法。第二类算法是体绘制算法,与等值面绘制方法相比,体绘制方法包含了原始数据中的每一个细节,不仅能显示高质量的图像,而且可以显示复杂的整个场结构及内部结构^[2],因此,体绘制研究深受人们的青睐。在体绘制的发展过程中,计算机硬件性能始终是一个关键的制约因素,尽管现在计算机的计算速度、内存容量以及其它图像加速设备的发展日新月异,但是对于庞大的体数据,体绘制速度始终难尽如人意。为了解决这种状况,研究人员们基于现有的硬件环境,根据各种应用的具体需求,在生成的图像质量和绘制速度之间寻求可能的最佳方案,使绘制速度尽量满足用户的要求,同时保证一定的图像质量^[3],因此,形成了许多不同的体绘制算法。比较常用的体绘制算法有:光线投射体绘制算法^[4]、剪切变形体绘制算法、三维纹理硬件支持直接体绘制算法等。

随着近年来 PC 机处理能力和图形显卡性能地飞速提高,原来只能在高档图形工作站上运行的体绘制算法现在可以移植到 PC 机上。然而,数据量庞大时,还是难以满足实际应用中的实时绘制和交互处理的需求。在加速算法中,基于硬件加速的方法主要有并行绘制方法、专用硬件方法及基于 PC 可编程图形处理器加速^[5]等。并行体绘制方法即通过高速网络互连的超级计算机间并行计算来获得高的加速比。大多数并行体绘制和专有硬件方法成本都比较高。相对于前两种方法高昂的成本而言,利用 PC 图形硬件加速方法具有易于实现、移植性好、性价比高、便于推广,技术进步快等优点。

2 纹理映射体绘制算法

基于纹理映射的体绘制是近年来出现的快速体绘制方法,它有效地利用图形硬件的绘制性能。纹理映射体绘制算法可分为两个步骤:第一步,纹理创建。体数据先由二维影像文件读入到系统内存的体缓存中,接着将体缓存中的数据转换为纹理数据并存储于纹理缓存中,然后纹理数据被装入图形硬件的纹理内存中并被定义为二维纹理或三维纹理。第二步,纹理绘制。定义一系列相互平行的多边形并通过纹理映射实现对体数据的重采样。最后混合纹理映射后的多边形生成结果图像。

纹理映射体绘制算法可分为二维纹理映射体绘制算法和三维纹理映射体绘制算法。其中基于二维纹理映射的体绘制方

基金项目:河南省自然科学基金(the Natural Science Foundation of Henan Province of China under Grant No.072300410180);河南省高校科技创新人才基金(No.2008HASTIT029)。

作者简介:许庆功(1973-),男,讲师,研究方向:图形图像处理及可视化技术;刘庆伟,讲师;张永胜,博士,教授。

收稿日期:2008-05-26 修回日期:2008-09-15

法是将体数据沿正交方向进行切片生成序列二维图像,产生二维纹理切片,然后将这些二维纹理切片依次堆叠来完成体绘制过程,其绘制方法简单,但绘制效果并不理想。目前应用比较广泛的是基于三维纹理映射的体绘制算法。

2.1 三维纹理映射体绘制算法

在描述三维纹理体绘制前,先介绍直接体绘制的光学模型。

对直接体绘制来说,基本思想是为屏幕上的每一个像素计算一个体绘制积分,如沿着每一条光线对颜色值和透明度值积分。假设光线上的一点 $X(\lambda)$ 是以其到视点的距离 λ 作为参数的,可以用 $S(X)$ 定义此点的数据值, $C(S(X))$ 定义此点的颜色值,用 $E(S(X))$ 定义此点的透明度值,则直接体绘制的光学模型积分为:

$$I = \int_0^D C(S(X(\lambda))) \exp\left(-\int_0^\lambda E(S(X(\lambda')))\right) d\lambda \quad (1)$$

由于用于体绘制的数据场是离散的三维网格状的,所以式(1)仍无法使用,需要对其近似的离散化。将一根光线 $X(\lambda)$ 等分为 n 段,每段长度为 $d=D/n$,将数据沿光线的分布曲线 $S(X(\lambda))$ 曲线用一条分段相等的曲线近似代替,如图 1 所示。

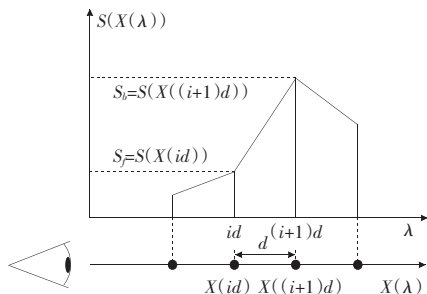


图 1 分布曲线

则 $\exp\left(-\int_0^\lambda E(S(X(\lambda')))\right) d\lambda$

可以被 $\exp\left(-\sum_{i=0}^{\lambda/d} E(S(X(id)))d\right) = \sum_{i=0}^{\lambda/d} \exp(-E(S(X(id)))d)$

近似地代替。

令 a_i, C_i 为第 i 段光线段上的数据所获得的不透明度和颜色值,则

$$\begin{aligned} a_i &\approx 1 - \exp(-E(S(X(id)))d) \\ C_i &\approx C(S(X(id)))d \end{aligned} \quad (2)$$

如此可将式(1)近似地写为:

$$I \approx \sum_{i=0}^n a_i C_i \prod_{j=0}^{i-1} \exp(1-a_j) \quad (3)$$

则式(3)也等价于

$$C_i' = a_i C_i + (1-a_i) C_{i-1}' \quad (4)$$

其中 C_i' 为光线通过第 i 段光线段所对应的数据点后的颜色情况。

设三维体数据的宽度、高度、深度分别为 W, H, D ,从视线方向将容积数据以单位距离 d 的间隔进行切片。其中距离 d 取决于容积数据单位体素的尺寸,根据绘制的质量来确定。设定颜色转换函数为 $C(\cdot)$,及阻光度转换函数为 $O(\cdot)$ 。

设光线从后向前经过纹理切片分别为 $S_1, S_2, S_3, \dots, S_{n-1}, S_n$,穿过其上的纹理点为 $P_1, P_2, P_3, \dots, P_{n-1}, P_n$,对应的颜色和阻光度值分别为:

$$C(f(P_1)), C(f(P_2)), C(f(P_3)), \dots, C(f(P_{n-1})), C(f(P_n))$$

$$E(f(P_1)), E(f(P_2)), E(f(P_3)), \dots, E(f(P_{n-1})), E(f(P_n))$$

依次经过所有纹理切片后,混合绘制结果为:

$$C_{final} = \sum_{i=1}^n C(f(P_i)) E(f(P_i)) - \sum_{i=1}^n \left(C(f(P_{n-i})) \prod_{k=1}^i E(f(P_k)) \right) \quad (5)$$

遍历所有光线就完成了对所有二维纹理切片的混合过程,如图 2 所示。

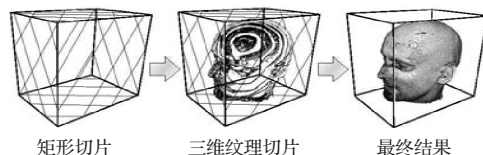


图 2 三维纹理绘制过程示意图

2.2 基于相关纹理的改进方法

三维纹理映射的体绘制方法利用图形处理器在切片上采样和混合,充分利用了现代图形硬件的像素并行处理、线性或三线性插值处理能力,具有极快的速度。但当切片数不是很大时,极容易出现伪影。从公式(1)可知, $C(S(X))$ 和 $E(S(X))$ 沿视线方向的采样频率近似为数据场 $S(X)$ 的奈魁斯特频率(Nyquist frequency)与传递函数 $C(S(X))$ 和 $E(S(X))$ 奈魁斯特频率最大值的乘积。传递函数的非线性特性显著增加了正确估算体绘制积分所需达到的采样频率,实际采样过程中往往无法满足,因而三维纹理体绘制方法所绘制的图像中往往会产生环状伪影。对此,对上面的方法进行改进:首先,在数据场中沿着光线采样,从采样到的点可获得一条分段线性曲线来近似代替数据沿光线的分布曲线 $S(X(\lambda))$ 曲线,见图 1 所示,令 $S_b = S(X(id))$, $S_f = S(X((i+1)d))$,则从第 i 段光线段所对应的数据点所获得的不透明度 a_i 为:

$$\begin{aligned} a_i &= 1 - \exp\left(-\int_d^{(i+1)d} E(S(X(\lambda)))d\lambda\right) \approx \\ &1 - \exp\left(-\int_0^1 E(S_f(1-\omega) + S_b\omega)d\omega\right) \end{aligned} \quad (6)$$

从第 i 段光线段所对应的数据点所获得的颜色 C_i 为:

$$\begin{aligned} C_i &\approx \int_0^1 C(S_f(1-\omega) + S_b\omega) \times \\ &\exp\left(-\int E(S_f(1-\omega') + S_b\omega')d\omega'\right) d\omega \end{aligned} \quad (7)$$

可见每一段从光线段所对应的数据点获得的颜色值与不透明度值只是 S_f, S_b 以及 d 的函数,其中 d 值属于预先设定, S_f, S_b 的所有可能组合都可以有穷的列举出来,因此 a_i 与 C_i 可以预先计算好放到查找表中,到时用 S_f, S_b 查表即可获得 a_i 与 C_i 的值。如此使从每段光线所对应的数据点中所获得信息量大增加,从而可以以比较小的采样率获得比较好的绘制结果。

实现该方法用 OpenGL 的 ARB_texture_shader 扩展库及相关纹理功能,结合 VC++7.0 实现。首先根据转换函数,预先计算出公式(6)和(7),把结果保存在一个二维纹理中,在绘制过程中,首先在三维体数据中沿光线采样,获得相应的 S_f, S_b ,再把 S_f, S_b 作为纹理坐标,在预先计算好的二维纹理中采样,得到对应的一段光线的颜色和阻光度值,最后根据光线混合公式进行计算,获得最终图像。实现步骤如图 3 所示。

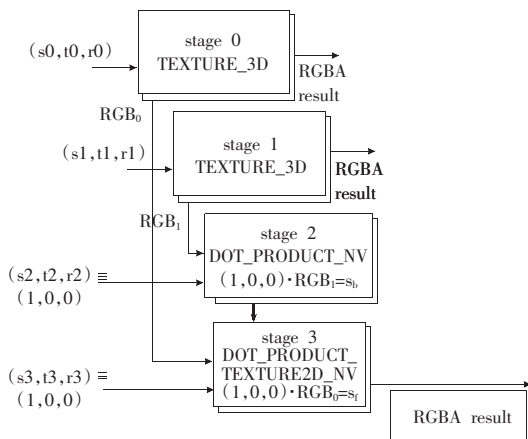


图3 相关纹理查找表实现过程

3 结论

在配置为 AMD2600+、512 M 内存、ATIX800 的 PC 机上，利用改进的相关纹理算法和已存的三维纹理映射体绘制算法绘制结果进行比较，如图 4 所示，数据如表 1 所示。对 128×128×128 的耳朵内部结构数据及 256×256×225 的头部体数据、256×256×280 胸腔体数据进行重建比较，其中显示窗口大小为 512×512。

由图与表可看出，相对三维纹理映射算法，算法的绘制速度相对较快，能达到更好的交互操作要求，绘制质量大幅度提高，消除了由于奈奎斯特采样频率高，而使三维纹理映射产生

表1 两种算法的实验结果比较

实验数据	数据大小	三维纹理映射(切片数)	相关纹理帧速率(切片数)
耳结构体数据	128×128×128	42.8 帧/s(128 slices)	27.3 帧/s(128 slices)
头部体数据	256×256×225	29.5 帧/s(256 slices)	18.9 帧/s(256 slices)
胸腔体数据	256×256×280	24.6 帧/s(256 slices)	13.7 帧/s(256 slices)

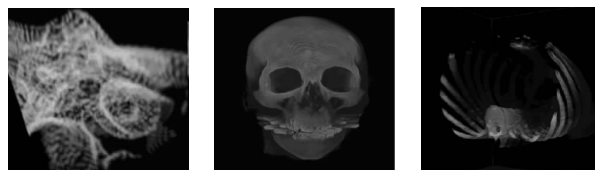


图4(a) 三维纹理映射算法绘制结果

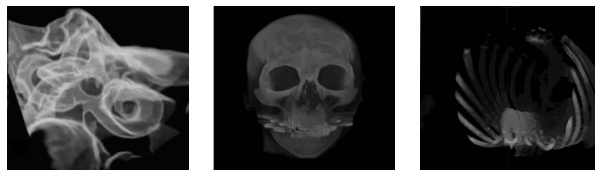


图4(b) 相关纹理算法绘制结果

的大量环状伪影，提高了绘图质量。虽然使用改进的算法能取得较好的绘制效果，但其所能处理的数据量还是受纹理内存的影响，当数据量大时，重建速度明显较慢。如何把相关纹理体绘制算法和空间跳跃算法结合，进一步提高运行速度和增加处理数据量，是后续的工作。

参考文献:

- [1] Lorensen W E, Cline H E. Marching cubes: A high resolution 3D surface construction[J]. Algorithm Computer Graphics, 1987, 21(4): 163-169.
- [2] 唐泽圣. 三维数据场可视化[M]. 北京: 清华大学出版社, 1999.
- [3] 田捷, 诸葛婴, 王靖, 等. 基于微机的三维医学图像处理与分析系统[J]. 中国生物医学工程学报, 2001, 20(3): 259-265.
- [4] 彭延军, 石教英. 体绘制技术在医学可视化中的新发展[J]. 中国图象图形学报: A 版, 2002, 12(7): 1239-1246.
- [5] 王少荣, 贾富仓, 孙晓鹏, 等. 数字人切片数据的硬件加速体绘制[J]. 计算机辅助设计与图形学学报, 2005, 17(9): 1998-2002.

(上接 175 页)

参考文献:

- [1] Boehm W, Farin G, Kahmann J. A survey of curve and surface methods[J]. Computer Aided Geometric Design, 1984, 1(1): 1-60.
- [2] Nielson G M. CAGD'S top ten: What to watch[J]. IEEE Computer Graphics and Application, 1993, 13(1): 35-37.
- [3] Piegl L. On NURBS: A survey[J]. IEEE Computer Graphics and Application, 1991, 11(5): 55-71.
- [4] Barsky B A. The β -spline: A local representation based on shape parameters and fundamental geometric measure[D]. Salt Lake: University of Utah, 1981.
- [5] Dierck P, Tjytgat B. Generating the Bézier point of β -spline curve[J]. Computer Aided Geometric Design, 1989, 6(2): 279-291.
- [6] Foley T A. Local control of interval tension using weighted splines[J]. Computer Aided Geometric Design, 1986, 3(2): 281-294.
- [7] Nielson G M. Rectangular v -splines[J]. IEEE Computer Graphics and Application, 1986, 6(1): 35-40.
- [8] Sarfraz M. Interpolatory rational cubic spline with biased point and interval tension[J]. Computer and Graphics, 1992, 16(4): 427-430.
- [9] Gregory J A, Sarfraz M, Yuen P K. Interactive curve design using C^2 rational splines[J]. Computer and Graphics, 1994, 18(2): 153-159.
- [10] Duan Qi, Djidjeli K, Price W G, et al. A rational cubic spline based on function values[J]. Computer and Graphics, 1998, 22(4): 479-486.
- [11] Duan Qi, Liu Ai-kui, Cheng Fu-hua. Constrained interpolation using rational cubic spline with linear denominators[J]. Korean Journal of Computational and Applied Mathematics, 1999, 6(1): 203-215.
- [12] Duan Qi, Djidjeli K, Price W G, et al. Some properties of a constrained rational cubic spline with linear denominator[J]. Korean Journal of Computational and Applied Mathematics, 2000, 7(2): 397-405.
- [13] 刘爱奎, 杜世田, 段奇, 等. 插值曲线的形状控制[J]. 工程图学学报, 2000, 21(1): 66-72.
- [14] Duan Qi, Djidjeli K, Price W G, et al. Weighted rational cubic spline interpolation and its application[J]. Journal of Computational and Applied Mathematics, 2000, 117: 121-135.
- [15] 刘爱奎, 段奇, 杜世田, 等. 插值曲线区域控制的加权有理插值方法[J]. 计算机辅助设计与图形学学报, 2000, 12(7): 497-501.
- [16] 刘爱奎, 段奇, 单沪军. 加权有理三次插值的逼近性质及其应用[J]. 高校应用数学学报: A 辑, 2000, 15(2): 211-218.
- [17] 谢楠, 张晓平. 一类有理三次样条的区域控制和逼近性质[J]. 山东大学学报: 工学版, 2004, 34(6): 106-111.
- [18] 邓四清, 方遼, 谢进. 一类有理三次插值样条曲线的区域控制[J]. 工程图学学报, 2008, 29(2): 104-109.
- [19] 邓四清, 方遼, 谢进, 等. 基于函数值的有理三次插值样条曲线的区域控制[J]. 计算数学, 2008, 30(2): 167-176.
- [20] 邓四清, 方遼, 谢进. 有理二次插值曲线的形状控制[J]. 计算机工程与应用, 2007, 43(29): 40-42.