文章编号:1000 - 582X(2005)07 - 0067 - 04

基于可编程 GPU 的快速体绘制技术*

张建勋,刘全利,陈 庄
(重庆工学院 计算机学院,重庆 400050)

摘 要:新一代的图形显示硬件集成了以图形处理器(Graphics Processing Unit,GPU)为核心的可编程顶点着色器和可编程像素着色器,为实现实时体绘制技术提供了硬件加速支持.该文首先分析了可编程 GPU 的绘制流水线、硬件体系结构和快速绘制原理.最后基于可编程 GPU 实现了医学体数据的快速最大密度投射体绘制方法.实验表明,采用 GPU 的可编程像素着色器进行体绘制所需要的时间明显地少与不用 GPU 的可编程像素着色器进行体绘制所需要的时间.

关键词:图形处理器;实时体绘制;最大密度投射;Cg
中图分类号:TP391.41 **文献标识码**:A

随着医学三维可视化技术的深入发展和应用,体绘制速度与用户要求实时、交互式处理之间的矛盾越来越严重.目前,进行三维医学数据集的快速绘制主要有两种方法:基于软件的加速方法和基于硬件的加速方法.

基于软件的加速方法中,国内外学者提出了如采用计算复杂度低的插值算法、降低重采样密度、发射更少量光线、采用简化的光照模型、梯度计算等方法来加速体绘制过程[1].但对于医学三维重建、手术导航等对绘制结果的准确性、精确性要求较高的应用,这些通过降低图像绘制质量来换取速度的方法是不可行的.另有一些更为科学的加速体绘制算法[2],如空间游程编码、八叉树等利用绘制的计算复杂度为 $O(N^3)$ 的特点,通过精心设计的数据结构有效地压缩和剔除那些对屏幕没有贡献的无效体素来提高体绘制速度.此外,提前不透明度截止、空间跳跃、提前计算视点无关数据等方法也用来对绘制算法进行加速处理.

基于硬件的加速方法主要有并行体绘制方法和采用专用硬件.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.大多数并行体绘制研究中,单独一台计算机上仍然只是实现软件算法,系统速度的提升还是过分依赖于单台处理机的能力;在能加速体绘制的专用硬件方面[3],德国 Tübingen 大学研究人员设计的 VOGUE 和 VIZARD Ⅱ 能够进行高质量的透视投影绘制,但系统需要高的内存和带宽、硬件加速效率不高.SGI 的 Texture Mapping 利用传统的多边形绘制算法中的纹理映射技术,在 SGI Reality Engine 4RM 上实现了对 256^3 规模体数据 10 帧/s 的绘制速度,主要缺点是系统需要巨大的带宽、绘制图像的质量较低.Mannheim 大学的 VIRIM 支持灵活的体绘制优化算法、阴影计算,但其性能提升受到全局总线的制约,对 256^3 规模体数据性能只能到 2.5 帧/s.上述所有系统都有一个共同的问题:价格昂贵(约 10 万美元)、缺乏灵活的传递函数设计(图像质量的关键).国内天津大学孙济洲授领导的研究小组也正在研究体绘制专用的 ASIC 图形卡[4].

随着现代图形硬件加速技术的进步,新一代的图形显示硬件集成了以 GPU(Graphics Processing Unit,图形处理器)为核心的可编程顶点着色器和可编程像素着色器,为实现实时体绘制技术提供了硬件支持.GPU 在 1999 年首先由 nVidia 公司提出,其发展速度是 CPU 速度的三倍多.目前图形芯片的主要市场被 nVidia 和 ATI 这两家公司占领.2001 年 GeForce3 出现,它是第一款可编程 GPU.最新的可编程 GPU 具有一些新的特征,概括起来有如下几方面[5]:1)在顶点级和像素级提供了灵活的可编程特性;2)在顶点级和

* 收稿日期:2005 - 03 - 09
基金项目:国家自然科学基金(60373061);重庆市自然科学基金(8647)
作者简介:张建勋(1971 -),男,四川夹江人,博士,重庆工学院副教授,研究方向为计算机图像图形、虚拟现实.

像素级运算上都支持 IEEE32 位浮点运算；3)支持多遍绘制的操作,这样避免了多次的 CPU 与 GPU 之间的数据交换；4)支持绘制到纹理的功能(Render - to - Texture/pbuffer),从而避免将计算结果拷贝到纹理这一比较费时的过程;5)支持依赖纹理功能,以方便数据的索引访问,可以将纹理数据作为已载入高速内存的数据来使用. 可编程 GPU 的出现,传统的内嵌固定绘制流水线(Fixed Function Rendering Pipeline)在向可编程绘制流水线(Programmable Rendering Pipeline)转变,使实现高度真实感图形/图像的实时体绘制更易实现. 随着可编程图形显示硬件的发展,在现今 OpenGL 扩展和 DirectX 图形 API 中均提供了直接操作 GPU 的编程接口,用户可以充分利用图形显示硬件的加速机制获取实时的真实感图形/图像绘制效果. 本文将详细阐述可编程 GPU 的硬件体系结构和快速绘制原理. 最后基于可编程 GPU 实现了医学体数据的快速最大密度投射(Maximum Intensity Projection,MIP)体绘制.

1 可编程 GPU 的绘制管线及硬件体系结构

1.1 可编程 GPU 的绘制管线

为了阐明基于 GPU 的体绘制技术,首先介绍图形显示硬件的可编程绘制流水线. 图形处理的流水线主要集中在顺序处理的两大部分:第 1 部分是对图元实施几何变换及对图元属性进行处理,即将几何模型的多边形/三角形顶点数据流从 CPU 交由图形处理部件实现几何变换及属性处理(包括部分光照计算). 第 2 部分则是在实现扫描转换进行光栅化以后进行一系列图形绘制处理,包括各种光照效果的合成、纹理映射、遮挡处理、反混淆处理等. 现代图形绘制流水线如图 1 所示,其中显著的特点是针对顶点光照计算增加了可编程顶点着色器(Vertex Shader),针对纹理加载和合成增加了可编程像素着色器(Pixel Shader). 它们都是典型的流处理机(stream processor). 这种流处理机与向量处理机的主要区别在于,它不具有大容量的快存/存储器可以读写,只是直接在芯片上利用临时寄存器作流数据的操作. 对于 GPU 而言,图形流数据分别是顶点图元及光栅化后的像素(理论上多个子素即 Fragment 组成一个像素). 图形显示硬件的 GPU 中分别提供了面向这两种着色器的机器指令和相应的寄存器,这些指令基本上以向量为操作数,在 GPU 支持下按照 SIMD 的并行方式执行. 目前在 Microsoft Direct 3D 和 OpenGL 中都提供了着色器的编程接口. OpenGL 中包括了 GPU 设计者(如 Nvidia)以及 OpenGL"架构委员会(ARB)"所扩充的函数. DirectX 则根据 GPU 新产品功能的扩充与进展及时地定义新的版本以扩充 Vertex Shader 和 Pixel Shader 的新功能. 直接使用 OpenGL 或 DirectX 扩充的接口软件可以从底层更灵活地控制和对 GPU 编程.

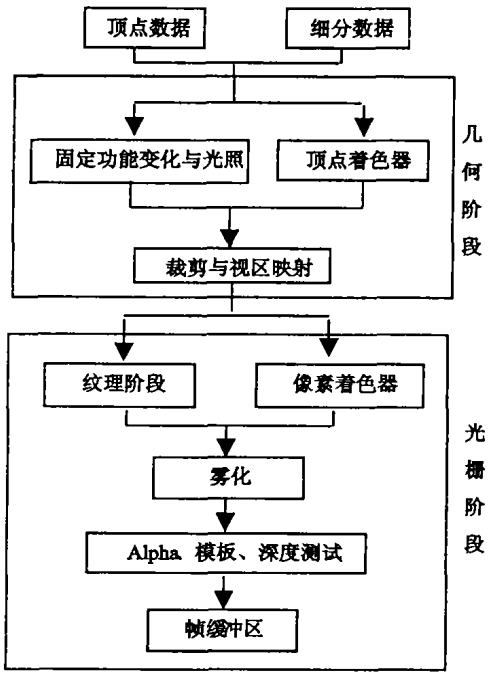


图 1 具有顶点着色器和像素着色器的管线

1.2 可编程 GPU 绘制的硬件体系结构

具有可编程 GPU 的图形卡可利用所提供的可编程硬件指令在纹理载入和 GPU 绘制期间实施光照计算. 如图 2 所示, GeForce3 中具有 4 层纹理绘制机制. 这 4 层纹理绘制机制在实施多重纹理载入的同时,设计了纹理层次之间的数学运算,这些数学运算主要包括纹理偏移运算和纹理点积运算(7 种不同类型的点积运算指令,如 DOT_PRODUCT_NV, DOT_PRODUCT_TEXTURE_2D_NV 等),借助这些硬件指令可以方便的构造纹理之间的点积运算,生成所需要的纹理表示. 一般实施点积运算的参数是归一化的向量表示,其取值范围为[0,1],为此在原有 GL_RGB 和 GL_RGBA 纹理格式的基础之上,扩展了 4 种新的 OpenGL 内部纹理格式,分别是 GL_SIGNED_HILO_NV, GL_HILO_NV, GL_SIGNED_RGB_NV 和 GL_SIGNED_RGBA_NV.

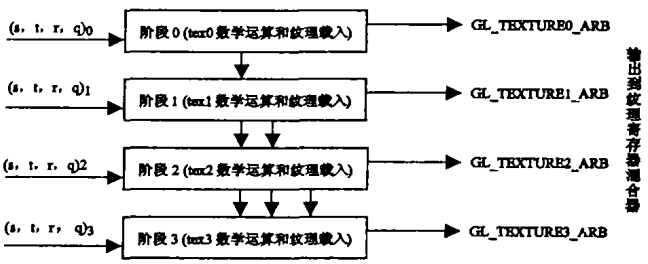


图 2 GeForce3 的纹理绘制处理逻辑流程图

另外,GeForce3 在纹理混合阶段提供层次化的纹理寄存器混合器,如图 3 所示,目前的 GeForce3 中具有 8 个普通纹理寄存器混合器和一个最终纹理寄存器混合器. 对于普通纹理寄存器混合器提供了 13 个内部寄存器,包括可读写的针对漫反射和镜面反射的 2 个颜色寄存器和 4 个纹理寄存器,2 个读写暂存寄存器,3 个只读寄存器,1 个雾化属性寄存器以及 1 个只写寄存器,这些寄存器混合器按照层次化处理模式排列,并且将 RGB 和 Alpha 混合并分别实施并行处理,每个混合器从寄存器组中可同时获取 4 个矢量化的 RGB 和 Alpha 纹理数据信息,经过向量化并行处理后将结果反送回寄存器组,供下一层混合器使用.

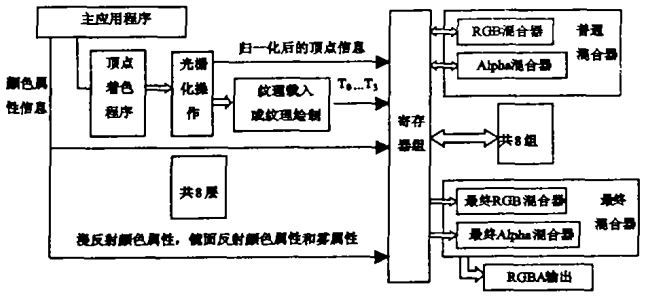
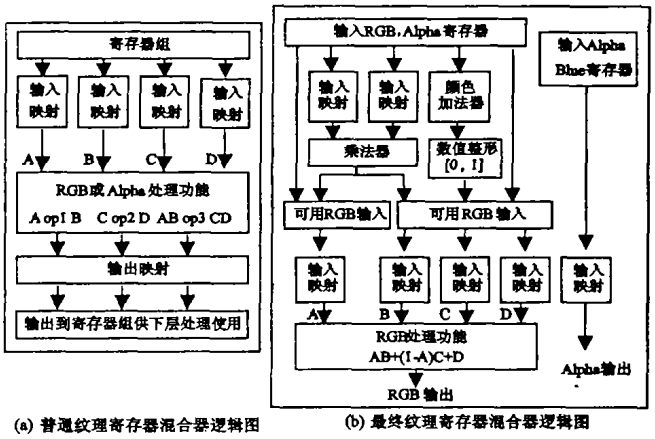


图 3　GeForce3 的纹理寄存器混合器工作流程图

对于每一个普通纹理寄存器混合器,硬件均提供了相应的处理功能和操作指令,其逻辑结构图如图 4(a)所示,其中输入映射主要针对输入纹理参数的整型变换操作指令,硬件中提供了 8 种映射模式和相应的操作指令,如指令 expand(tex0)实施 f(x) = 2 * max(0, 1) - 1 即[0,1]→[-1,1]纹理参数变换,混合器中 RGB 和 Alpha 的处理功能是分开的,3 个输出通道可分别实施 op 操作,其中 op = {DOT, Mult, Mux, Sum},分别为点积、乘法、选择和求和运算,而 Alpha 通道的 3 个输出分别为 A * B, C * D 和 A * B(op) C * D,输出映射同样实施数据变换,硬件中分别设计了相应的比例和偏置指令,变换的结果供下一级混合器处理和送入最终混合器处理,值得注意的是这 8 层普通混合器是可选的,用户按照自己算法设计的需要使用混合器资源. 普通混合器处理完成之后,必须将数据送入最终混合器处理之后方能形成最后的 RGBA 输出到下一级图形绘制流水线,最终混合器的逻辑结构图如图 4(b)所示,其工作模式基本与普通混合器相似,不同的是它有 6 路 RGB 输入和 1 路 Alpha 输入,并且没有点积运算,其目的是实现颜色的最终融合,它的 RGB 通道的处理功能固定的,即输出只能是 A * B + (1 - A) * C + D,且 A 的输入不能是颜色求和的结果.



(a) 普通纹理寄存器混合器逻辑图　　(b) 最终纹理寄存器混合器逻辑图

图 4　寄存器混合器逻辑图

3　基于可编程 GPU 的体绘制技术

可编程 GPU 绘制管线及硬件体系结构的设计和分析的主要目标就是如何将医学影像数据(三维体素集)的三维绘制过程分解成为不同的纹理表示,然后通过硬件提供的多道纹理技术(MultiTextureing)、纹理绘制技术(Texture Shading)和纹理寄存器混合器(Register Combiner)实现像素级的实时绘制. 其中,顶点着色器操作在空间的几何点(单个体素)上,适用于几何操作类的应用,如视点的改变. 而像素着色器由于是操作在像素一级的单元上,具有较大容量的"纹理"空间,可以模拟纹理数据及其操作,如选择、融合、剪切等操作.

3.1　体绘制技术

近十年来体绘制技术在算法研究方面取得了巨大进展,先后提出了体光线投射、足迹表法(Splatting)、体元投射法(Cell Projection)、错切 - 变形(Shear - Wrap)、频域体绘制(Frequency Domain Volume Rendering)、以及基于纹理映射(Texture Mapping)的体绘制、最大密度投射(MIP)等众多算法[6]. 最大密度投射实际上不是真正意义上的融合计算,而是直接将一条光线上各个重采样点中的最大密度值当作该光线或者对应屏幕像素的颜色. 最大密度投射主要应用于医学影像领域,尤其是在对核磁共振血管造影术(Magnetic Resonance Angiography, MRA)数据进行血管造影中得到广泛应用.

3.2　最大密度投射算法的实现

Nvidia 公司的 Cg 是最早为可编程图形硬件设计的高级编程语言,为用户提供了直接基于 API(OpenGL 或 DirectX)编程的较为方便和高层次的工具. 以下是实现最大密度投射的 pixel shader 的 Cg 原代码.

```
struct PIN { float3 coord3d : TEXCOORD0;};
float4 main(const sampler3D in uniform pix3dtex : TEX-
TURE0,          //pix3dtex 存放由体数据组成的 3D
//纹理
const sampler2D in uniform pixLUT,          //pix-
LUT 存放 2D 传递函数(映射密度值到各颜色通道)
const float3 in uniform pixModifier,          //pix
//Modifier 存放最大密度值
const PIN in pin) : COLOR0
{
uniform float4 origColor = tex3D ( pix3dtex, pin.
coord3d); //用原始密度值计算子素的颜色,此处还
//未使用传递函数
uniform float4 surfColor;          //surfColor
//存放子素的颜色和不透明度
uniform float modColor = tex2D(pixLUT, float2(0,orig-
Color.w)).w;          //计算第 4 个颜色通道(Alpha
//通道)密度值
surfColor.x = tex2D(pixLUT, float2(0,origColor.x)).
x + pixModifier.x * modColor;//计算子素的颜色和不
//透明度
surfColor.y = tex2D(pixLUT, float2(0,origColor.y)).
y + pixModifier.y * modColor;
surfColor.z = tex2D(pixLUT, float2(0,origColor.z)).z
+ pixModifier.z * modColor;
surfColor.w = max(surfColor.x, max(surfColor.y, sur-
fColor.z));
return surfColor;
}
```

表 1 算法性能比较表

实验数据	分辨率	不用 pixel shader 的绘制时间/s	用 pixel shader 的绘制时间/s
腹腔 CT1	512×512×60	2.27	0.41
腹腔 CT2	512×512×106	5.38	0.45

图 5 是采用最大密度投射法由病人腹腔 CT 扫描切片重构的 3 维图像,所采用的实验平台是 2.0G 奔 4 CPU 的 PC 机,内存为 256 M,GeForceFX 5200 显卡,显存为 128 M. 图像的分辨率为 800×600. 不用 pixel shader 进行体绘制时,沿每条光线的采样间隔就是切片间的间距. 而采用 pixel shader 进行体绘制时,3D 纹理的多边形采用 106 个多边形.

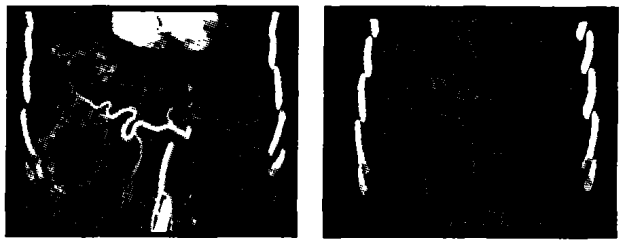


图 5 采用最大密度投射法由病人腹腔 CT 扫描切片重构的 3 维图像

4 结 论

对于规模为 512×512×100 分辨率(这是医学检验常用的数据量)的规则体数据,要求以 15 Hz 的刷新频率,显示分辨率为 800×600 的医学三维实时可视化系统目前还没有在 PC 机上实现. 在速度无法保证实时的情况下,交互性也就无从谈起,虚拟手术、手术导航等应用更无从谈起. 在保证绘制质量的前提下,提高速度成为当前制约医学三维可视化技术及其进一步发展、应用的主要难题. 而目前在单 PC 机上实现具有高度真实感的实时医学可视化系统,唯一可行的方法是利用高端图形卡的 GPU 可编程性编程实现. 因此,研究基于可编程 GPU 的高度真实感的实时医学可视化系统具有非常重要的意义. 随着更新、更强大的带可编程性 GPU 的高端图形卡的面市,具有高度真实感的实时医学可视化系统很快就能实现.

参考文献:

[1] 唐泽圣. 三维数据场可视化[M]. 北京:清华大学出版社, 1999.

[2] 张加万. 交互式体绘制关键技术及其应用研究[D]. 天津:天津大学, 2004.

[3] MARKUS HADWIGER,JOE M KNISS,KLAUS ENGEL. High-quality Volume Graphics on Consumer PC Hardware[Z]. Siggraph'2002.230 - 274.

[4] ZHANG JIAWAN, SUN ZHIGANG, SUN JIZHOU. VolGraph: A PC-based Visualization System for Three-dimensional Medical Reconstructions[J]. Transactions of Tianjin University (English Version), 2003, 9(3): 206 - 210.

[5] 吴恩华. 图形处理器用于通用计算的技术现状及其挑战[J]. 软件学报,2004, 15(10):1 493 - 1 504.

[6] MICHAEL MACEDONIA. GPU Enters Computing's Mainstream [J]. IEEE Computer Society, 2003, 36(10):106 - 108.

(下转第 77 页)

Fuzzy Clustering Theory for Analyzing Intrusion Detection Data

XIAN Ji-qing[1], LANG Feng-hua[2]

(1. School of Automation, Chongqing University of Posts and Telecommunication, Chongqing 400065, China;
2. School of Computer Science and Technology, Chongqing University of Posts & Telecommunication, Chongqing 400065, China)

Abstract: Intrusion detection system is an important component of the computer and information security framework. Its main goal is to differentiate between normal activities of the system and behaviors that can be classified as suspicious or intrusive, and its main challenge is to efficiently detect intrusion detection behaviors for reducing false positive rate and false negative rate. In view of the disadvantages of the existing intrusion detection methods, fuzzy c-means(FCM) clustering method is used to analyze intrusion detection data in order to detect anomaly network behavior patterns. Experimental results on the CUP99 data set data show that this method can not only feasible but also improve the accuracy and efficiency.

Key words: intrusion detection; anomaly detection; fuzzy clustering; fuzzy c-means clustering

(编辑　吕赛英)

(上接第 70 页)

Fast Volume Rendering Technology Based on Programmable GPU

ZHANG Jian-xun, LIU Quan-li, CHEN Zhuang

(School of Computer Science & Engineering, Chongqing Institute of Technology, Chongqing 400050, China)

Abstract: Techniques of programmable vertex shader and pixel shader have been integrated in newly developed graphics hardware armed with powerful Graphics Processing Unit(GPU) in recent years, and as a result, real-time volume rendering can be implement. First, rendering pipeline, hardware architectures on per-pixel shading and fast rendering principium of the modern programmable GPU are explained in detail. Second, technology on how to analyze and solve volume rendering problems is described. Finally, maximum intensity projection (MIP) method rendering medical volume data have been implemented based on programmable Graphics Processing Unit. In a performance test, spent time rendering medical volume data based on programmable pixel shader in GPU is obviously less than spent time rendering it do without programmable pixel shader.

Key words: graphics processing unit; real-time volume rendering; maximum intensity projection; Cg

(编辑　张小强)