文章编号:1671-8860(2010)02-0143-04　　**文献标志码**:A

利用可编程 GPU 硬件进行大规模真实感地形绘制

靳海亮[1,2,3]　卢小平[1]　刘慧杰[1]

(1　河南理工大学矿山空间信息技术国家测绘局重点实验室,焦作市高新区世纪大道 2001 号,454003)
(2　大连大学辽宁省智能信息处理重点实验室,大连市经济技术开发区学府大街 10 号,116622)
(3　江西省数字国土重点实验室,抚州市学府路 54 号,344000)

摘　要:提出了一个用于大地形数据外存可视化的硬件加速的视点相关 LOD 渲染算法。为了在可视化中实现真实感,把相应的卫星或航空图像纹理应用在地形上,还提出了在保持实时帧速的同时,在地形上显示如建筑物、树木、道路等大量的静态物体集合的高效率的目标处理算法。上述算法已成功地在不同的实际地形数据和卫星影像纹理上进行了测试,测试结果表明,基于可编程 GPU 硬件的大规模地形绘制速度快、真实感强。

关键词:大规模地形;可编程 GPU 硬件;真实感;绘制
中图法分类号:P208

实时地形可视化中最大的问题是如何管理大型数据集,包括几何形状(高度图)和纹理信息。LOD 模型通过在复杂性和准确性之间进行优化折衷来表示和处理地形数据。但是,目前的 LOD 算法[1]是高度 CPU 密集型的,并且几何吞吐量对图形硬件来说仍然是一个瓶颈。为了在现有的硬件水平上实现地形模型的快速绘制,自 1976 年 Clark 提出由模型简化来构造多分辨率模型[2]的思想以来,各类 LOD 模型的建立已成为提高地形绘制速度的重要方法之一,而硬件上则主要依赖于图形卡厂商显卡性能的提高。此外,该领域也有一些 CPU 多媒体指令和并行计算方法的应用研究。

近年来,GPU 的出现给复杂图形的快速绘制注入了新的活力,而高级 GPU 编程语言的出现则使基于 GPU 编程的应用开始普及起来[3-5]。本文利用当前 GPU 可编程图形处理流水线的特点,在可视化中通过发展合适的渲染程序来实现高级的视觉效果(如雾)和真实感(如用于细节纹理的多重纹理)的方法,并用大型地形数据集对所提出的算法进行了测试。

1　基于可编程 GPU 的真实感地形绘制

1.1　海量地形和对象数据实时绘制

1.1.1　硬件加速的大地形 LOD 渲染技术

运用多分辨率细节层次网格简化方案,近的区域比远的区域细节更高,使得最终产生的图像没有任何明显的视觉差别。本文采用基于块的视相关动态 LOD 模型[6-8]用于网格简化,并在实时渲染中使用相关细节纹理来显示高分辨率卫星图像细节。地形几何数据和纹理数据分别被组织成 $(2n+1)\times(2n+1)$ 和 $2n\times 2n$ 大小的片($n=8$),在相邻片之间保持一个像素的重叠,以确保几何片之间正确地缝合。在任一时间,只有 9 个片的数据存放在主存储器,假定观察者在中心片内。此外,高度图和纹理图像数据被预处理并存储为多分辨率形式,以支持飞行漫游模式。在步行漫游模式下,系统选择分辨率最高的数据;而在飞行

收稿日期:2009-12-17。
项目来源:国家重点基础研究发展计划资助项目(2009CB226107);矿山空间信息技术国家测绘局重点实验室开放研究基金资助项目(KLM200912);河南理工大学博士基金资助项目(B2009-23);河南省科技发展计划重点攻关资助项目(092102210362);河南省教育厅自然科学研究计划资助项目(2010A420003);辽宁省智能信息处理重点实验室开放课题资助项目(2007-1);江西省数字国土重点实验室开放研究基金资助项目(DLLJ200604)。

漫游模式下,系统基于观察者离地面的高度选择适当的分辨率数据。

为加快视景体裁剪,每个几何片的数据都使用一个带有叶子的四叉树进行组织,叶子对应$(2m+1)\times(2m+1)$大小的块或片($m=4$)。算法不处理单个三角形,而是处理几何簇(称为块的三角形集合),这样,CPU要执行的工作量就大为减少。鉴于对每个片进行多分辨率表示,顾及地形的复杂性以及观察者飞越地形时观察者的位置,算法使用一个变量τ(屏幕空间阈值)来限制投影影像的最大误差。此外,为LOD块构造长三角形条带解决了CPU到显卡的带宽问题,并可避免多余的三维顶点转换和光照计算。

1.1.2 数字地形上的对象管理

要显示物体的位置点,使用分页技术和对象实例化的对象处理方法。这里考虑了两类静态对象,第一类是简单的对象,具有简单的几何形状,能在OpenGL原函数的帮助下进行绘制,这些对象不需要加载到内存中。第二类是复杂的对象(使用AutoCAD三维模型绘制),具有复杂的几何形状和大量的三角形,这些对象包括它们的顶点和拓扑信息需要加载到主存储器中。

三维多分辨率地形上折线矢量数据的显示采用基于几何的映射方法实现。对于矢量数据的多分辨率建模,算法允许系统适应视觉映射(不向上下文渲染人造地物)和保持交互帧速时的用户需求。

1.2 可编程图形流水线技术

近年来,图形硬件性能比处理器性能提高得更迅速[9]。现代的GPU提供顶点级和像素级的编程处理,这种性能允许在用户级硬件上渲染增强真实感和细致的视觉效果,超过传统固定功能的渲染流水线可以提供的效果。在一个固定功能的渲染流水线条件下,数据能传给流水线,也能设置流水线状态,但是不能指定点和片元处理阶段的直接变换。一个常见的例子是许多三维渲染流水线所使用的光照模型。DirectX和OpenGL都使用基于Phong的光照公式,因为这个公式在每个顶点上都能很容易地计算。尽管存在诸如全局光照和BRFD等很多光照模型[10],但是开发者都受限于这一有效的模型,类似的限制例子遍及整个流水线过程。

可编程流水线允许通过创建着色器完全控制流水线的特定阶段,编译过的代码小片断取代特定阶段提供执行的标准供应商代码。顶点着色器是执行顶点和法向转换、纹理坐标生成以及顶点光照计算的程序,通常在几何处理阶段计算。片段或像素着色器是在图形流水线的像素处理阶段执行计算的程序,精确确定每个像素如何着色、纹理如何应用和一个像素是否应该绘制或不绘制。

顶点程序是在图元组装之前在顶点上执行,片断程序是在光栅化之后执行。这些小着色程序从用户程序传送到图形硬件(见图1),并在图形硬件上执行。

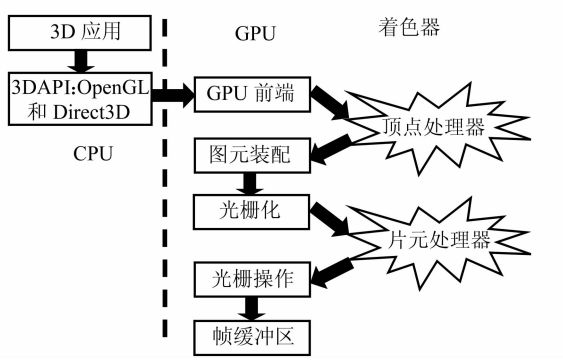


图1 可编程图形流水线

Fig. 1 Programmable Graphics Pipeline

1.3 使用GPU编程增加视觉效果和真实感

到目前为止,大多数GPU程序都是用汇编语言编写的。随着可用的汇编命令和功能的增加,容易看到,随着着色器持续增长的规模和复杂性,类汇编语言是不适当的。目前,在实时3D使用中,有两个主要的图形API:DirectX和OpenGL,每个API都提供一个高级着色器语言,分别为HLSL和GLSL。图形硬件供应商NVIDIA公司提供了第三个Cg(图形C语言),可以使用两个API中的任何一个。这些高级着色器语言都是围绕与C语言的子集非常相似的语法和熟悉的框架设计的。然而,与传统的编程语言相反,上述着色器语言是基于数据流计算模型,并显式地利用GPU的流特点的(单指令多数据或SIMD控制)。

这种交叉的API支持是通过翻译配置文件来完成的,配置文件允许着色器把各种各样的硬件作为目标,包括标准的DirectX着色器等级、OpenGL的着色扩展和特定的NVIDIA硬件特性等。当一个着色器发送到转换器,用户指定一个配置文件标志,转换器将生成运行于硬件的代码。

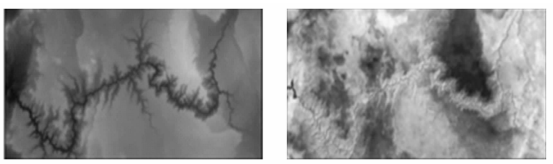
以顶点和片元程序为基础,笔者探索了基于Cg的GPU程序设计方法[11],并与OpenGL的3D API和V C^{++}代码集成起来(这些代码实际上是在计算机的GPU内执行的),用于多重纹理映射以及雾效果。GPU的编程代码由三部分组

成:初始化与设置、顶点程序和片元程序。第一部分初始化变量,在主程序中的变量和着色器中的变量之间提供一个联系。顶点程序计算统一的 fogDensity 参数,根据到眼睛的最短距离计算一个雾指数。然后,片元程序采样一个贴花纹理,用插值颜色调制混合贴花颜色,并雾化纹理片元的颜色(假设一个插值雾指数和一个统一的雾色)。

2　结果及性能分析

在 Win32 环境下,用 VC^{++} 结合 OpenGL 的 3D API 开发了一个三维地形实时可视化系统 3D Terrain,使用 Cg 语言进行 GPU 编程,并用大峡谷 4 K×2 K 的地形数据集和普吉特湾地区 16 K×16 K 的地形数据集进行了测试。

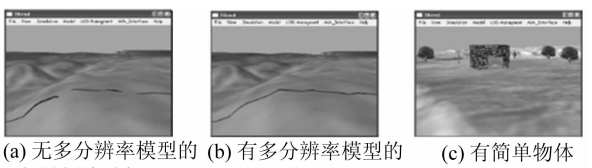
图 2(a)和图 2(b)分别显示了大峡谷区域的 DEM 和相应的卫星图像。图 3(a)显示在没有折线矢量数据多分辨率模型的 LOD 地形上叠置折线矢量数据,图 3(b)显示在有折线矢量数据多分辨率模型的 LOD 地形上叠置折线矢量数据,图 3(c)显示在三维地形上叠加简单物体,图 3(d)显示在三维地形上叠加一个复杂物体,图 3(e)显示详细的草地纹理,图 3(f)显示在场景中增加真实感的雾效果和多重纹理(用来混合草地纹理和卫星图像纹理)。



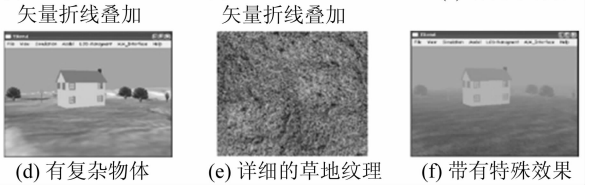
(a) DEM(高度图)　　(b) 相应的卫星图像

图 2　大峡谷区域的地形栅格数据(大小为 4 K×2 K)

Fig. 2　Terrain Raster Data of Grand Canyon Area



(a) 无多分辨率模型的矢量折线叠加　(b) 有多分辨率模型的矢量折线叠加　(c) 有简单物体



(d) 有复杂物体　(e) 详细的草地纹理　(f) 带有特殊效果

图 3　矢量折线和基于对象的点位显示

Fig. 3　Display of Vector Polyline and Point-location Based Objects

笔者对软件的性能进行了测试,计算机配置为 Pentium Ⅳ 2.4 GHz CPU,512 M 内存,显卡为 GeForce 4Ti4600,128 M 显存。用于栅格数据集三维可视化的自适应 LOD 算法性能分析如表 1 所示,相同的自适应 LOD 算法(使用三角形条带(用索引顶点数组))连同地物对象管理算法的测试结果如表 2 所示。此外,还用 Nvidia 公司的讯景 GeForce 6800GT 256 M 可编程图形卡来测试顶点和片元程序,这些程序通过结合多重纹理和雾效果的特点来增加视觉效果和真实感。

表 1　自适应 LOD 算法性能分析(无物体对象和矢量数据)

Tab. 1　Performance Analysis of the Adaptive LOD Algorithm(Without Objects and Vector Data)

不带地物对象的地形渲染		平均三角形数	平均帧速率/帧·s^{-1}
全分辨率(仅考虑 9 个地形块)		1 327 104.00	1.72
视域剔除		268 120.70	7.18
采用 LOD 算法(τ=4)	①使用三角形列表	20 368.45	57.23
	②使用三角形扇	20 368.44	74.11
	③使用三角形条带(不使用索引顶点数组)	23 655.06	114.33
	④使用三角形条带(使用索引顶点数组)	23 733.79	130.79

表 2　自适应 LOD 算法性能分析(有物体对象和矢量数据)

Tab. 2　Performance Analysis of the Adaptive LOD Algorithm (with Objects and Vector Data)

带地物对象的地形渲染	平均帧速率/帧·s^{-1}
① 使用 758 个地物对象(OpenGL:39 个,布告板:719 个)	117.17
② 使用 763 个地物对象(复杂对象:5 个,OpenGL:39 个,布告板:719 个)	111.32
③ 使用多分辨率矢量线(共 573 个点)和 758 个地物对象(OpenGL:39 个,布告板:719 个)	85.16

3　结　语

为了在普通的台式机上实现大规模真实感的地形模拟,且渲染必须做到实时,以确保帧速 30 帧/s 以上,笔者实现了一个高效的基于片的外存视相关 LOD 网格简化算法,在地形上显示诸如建筑物、树木、路等大量的静态物体集合,同时保持实时帧速,用有效的对象处理算法进行增强。为了在三维地形可视化中实现真实的视觉效果,通过开发合适的顶点和片元着色程序来利用当前 GPU 的可编程图形流水线的功能。通过 GPU 编程以及考虑到 GPU 流数据处理模型的性能效率,把一些通用的计算从 CPU 下移到 GPU 上。GPU 上的可编程数据处理是图形学、三维真实感

可视化的未来发展方向。

参 考 文 献

[1] Hesse M, Gavrilova M L. An Efficient Algorithm for Real-Time 3D Terrain Walkthrough[J]. International Journal of CAD/CAM, 2003,3(2):111-117

[2] Clark J H. Hierarchical Geometric Models for Visible Surface Algorithm[J]. Communications of the ACM, 1976,19(10): 547-554

[3] Losasso F, HoppeH. Geometry Clipmaps: Terrain RenderingUsing Nested Regular Grids[C]. ACM SIGGRAPH, LosAngeles, California, USA, 2004

[4] Schneider J, Westermann R. GPU-friendly High-quality Terrain Rendering[J]. Journal of WSCG, 2006,14(1): 49-56

[5] Malte C, Christian H H. Terrain Rendering Using Spherical Clipmaps[C]. IEEE Conference on Visualization, Lisbon, Portuga, 2006

[6] 杜剑侠,李凤霞,战守义.基于外存的大规模地形可视化框架[J].昆明理工大学学报(理工版),2006,31(5):1-5,13

[7] 蒲浩,宋占峰. 基于可见性预处理的地形模型视相关简化算法[J]. 武汉大学学报·信息科学版,2005,30(7):636-639

[8] 许妙忠. 大规模地形实时绘制的算法研究[J]. 武汉大学学报·信息科学版,2005,30(5):392-395

[9] 康宁,徐青,周杨,等.一种基于图形硬件的海量地形实时可视化算法[J].系统仿真学报,2007,19(17):3 988-3 992

[10] Engel W. ShaderX3: Advanced Rendering with DirectX and OpenGL[M]. Handcover: Charles River Media Graphics, 2004

[11] Fernando R, Kilgard M J. The Cg Tutorial: The Definitive Guide to Programmable Real-Time Graphics[OL]. http://www.jsj.tzc.edu.cn/teacher/crq/chmjiaochen/game/The%20Cg%20Tutorial%20-%20The%20Definitive%20Guide%20to%20Programmable%20Real-Time%20Graphics.pdf,2003

第一作者简介:靳海亮,博士,副教授,现主要从事GIS与数字城市、地形三维可视化等研究。

E-mail:jin_hailiang@126.com

Large-Scale Terrain Realistic Rendering Based on Programmable GPU Hardware

JIN Hailiang[1,2,3] *LU Xiaoping*[1] *LIU Huijie*[1]

(1 Key Laboratory of Mine Spatial Information Technologies of State Bureau of Surveying and Mapping, Henan Polytechnic University, 2001 Shiji Road, Jiaozuo 454003, China)

(2 Liaoning Key Laboratory of Intelligent Information Processing, Dalian University, 10 Xuefu Street, Dalian 116622, China)

(3 Digital Land Key Laboratory of Jiangxi Province, 54 Xuefu Road, Wuzhou 344000, China)

Abstract: An efficient hardware-accelerated view-dependent level-of-detail rendering technique for out-of-core visualization of large terrains is developed. To achieve realistic in visualization, the corresponding satellite or aerial imagery texture is applied over the terrains. Also, to display large collection of static objects such as buildings, trees, roads, and so on. Over the terrain while maintaining the real-time frame rates, efficient object handling methods are developed. A technique to use powerful performance and programmable vertex and fragment features of current GPU for advanced visual effects and increased realism in terrain visualization are proposed. The above algorithms have been successfully tested on different terrains and satellite images. Results show that large-scale terrain renders fast and with high realistic using GPU-based programmable hardware.

Key words: large-scale terrain; programmable GPU hardware; realistic; rendering

About the first author: JIN Hailiang, Ph.D, associate professor. His research interests are GIS and digital city, 3D terrain visualization, etc. E-mail: jin_hailiang@126.com