

基于CUDA的体绘制GPU加速算法

吴磊 王彬

东北大学中荷生物医学与信息工程学院, 沈阳 110004

摘要 本文主要是针对传统意义上的体绘制的经典算法——光线投射算法,在已有的GPU加速的基础上,提出了利用CUDA实现对算法的优化和加速。具体来讲,首先需要将体数据以三维纹理的形式储存在GPU的显存中,之后利用CUDA的并行处理能力对显存中的数据进行相应的处理,主要包括:生成存储顶点灰度值和颜色的纹理数组,计算顶点梯度,实现坐标系的转换,确定有效光线,利用Phong光照模型来求得光线上各等距采样点的反射分量,最后利用累加函数求得屏幕上每一点的像素值。实验结果表明:我们利用CUDA对光线投射算法的实现比传统的GPU加速算法大大提高了速度和成像质量。

关键词 光线投射、三维纹理、三维数据场、Phong光照模型、数据重建

中图分类号 TP391.41 文献标志码 A

引言

目前在医疗领域,三维数据的可视化具有十分重要的意义,而当前我们对大规模的体数据集进行形象描述的具体方法主要有面绘制和体绘制两大方面,体绘制是目前三维重建的主流,但是由于各方面因素的影响,绘制速度一直是限制体绘制发展的瓶颈。

近年来随着GPU的发展,一些基于GPU的加速算法得以实现,其的确大大加快了体绘制的速度,但离实时性还是有一段距离,并且传统的GPU加速本身也存在着一些不可避免的缺点。2006年Nvidia公司提出了CUDA体系架构,从根本上为GPU编程提供了一个全新的软硬件架构。首先它使GPU编程摆脱了以往的必须具有大量的图形应用接口的案例,它提供了专门的硬件访问接口,这对于广大编程者来说,利用CUDA来实现GPU编程比传统意义上的GPU编程大大降低了难度,提高了效率。其次CUDA在架构上采用了一种全新的计算体系结构来使用GPU提供的硬件资源,线程、线程块和网格的提出给大规模的数据处理提供了一种比CPU更具效率的方式。最后,CUDA集成了CPU和GPU的共同开发,其中内核部分为在GPU上执行的部分。由此可见,如果能将CUDA应用于医学领域的三维体数据绘制来作为新一代的GPU加速算法,必将在原始的基础上大大提高运算的效率和执行速度。

1 光线投射算法

经典的光线投射算法的主要原理是首先向整个数

据场发射光线,根据光线穿透体数据场的方式分为:平行光透射和视点光透射,平行光透射又叫正交投影,视点光透射因为与人眼的观察方式相似,因而看起来更具真实性。

当光线遍历体数据集后,根据设定的原则对数据场间隔均匀的采样,并赋予相应点对应的颜色值,然后利用积分的原则对每一条上的点值进行累加最后将得到的值作为屏幕上该点像素的值,最终形成图像得以显示。

由于光线投射算法需要遍历整个体数据集,因而早期的该算法效率很低,对此人们提出了一系列的改进措施,文献^[3]提出了利用GPU来实现八叉树的体绘制减少了计算量,文献^[4]实现了在GPU上对三维数据集的遍历、采样、计算的整个过程操作,加快了计算速度,但是由于传统GPU计算上存在的固有的缺点,使这些加速算法在提高速度的同时依然具有本身难以克服的缺点。

2 利用CUDA实现光线投射算法

2.1 CUDA的架构

CUDA体系通过对GPU硬件资源的全新分配,实现了GPU计算的并行能力。CUDA的硬件体系结构设计为由一系列“单指令,多数据(SIMD)”的多处理器组成,每一个多处理器有一个共享的内存、固定的缓存、纹理缓存,同时由多个处理器组成,(见图1)在任意时刻,处于同一多处理器中的处理器将执行同一条指令。

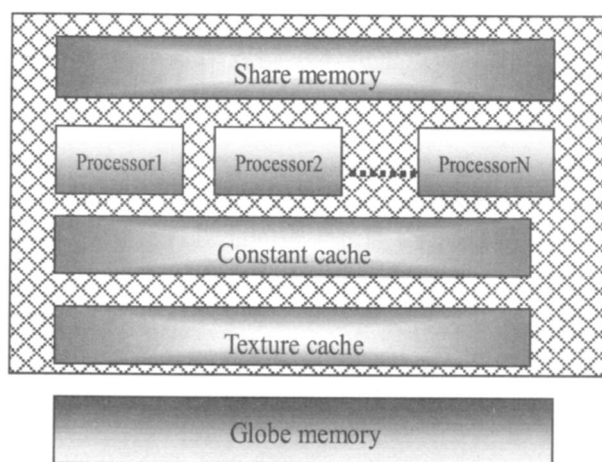


图 1 CUDA 的硬件体系架构

基于 CUDA 的程序设计以 C 语言实现,从而代替了原来 GPU 编程中必须加入大量图形应用接口语言如 OpenGL 等等所带来的不便,使 GPU 编程开始越来越灵活。另外在 CUDA 体系中将 GPU 设计成了能够同时执行大量线程操作的计算设备 (Device),大量的线程以网格的形式被组成了线程块,并且处于同一个线程块的线程能够同时并且高效的借助一个共享内存来共享数据。这样处于同一个线程块中线程其实就实现了数据的并行处理,即所有的线程共同执行一段代码 (“单指令,多线程”架构,缩写为 SIMT),同时当主机 (host) CPU 上的 CUDA 程序调用内核网格时,网格的线程块将被枚举并分发到具有可用执行容量的多处理器上。一个线程块上的线程在一个多处理器上并发执行,并且每个线程都有自己的一个 ID,便于计算和管理。线程块终止时,将在空闲多处理器上启动新块。(原理见图 2)

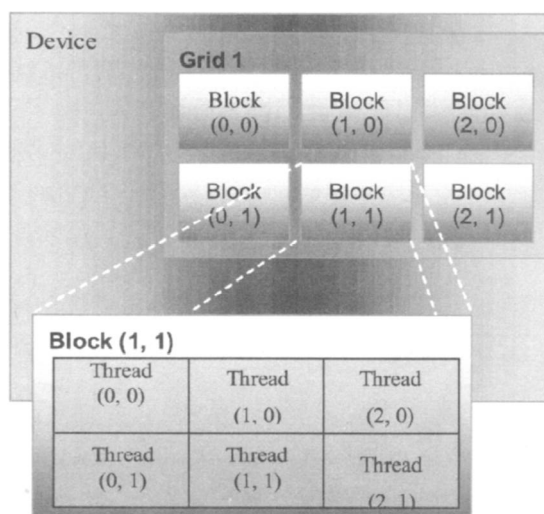


图 2 基于 CUDA 程序设计基本架构

2.2 具体实现

现在,如果我们要利用 CUDA 来实现光线投射算法,我们只需要声明一个内核函数来进行体数据遍历、光线方向上的等距采样以及颜色值的累加计算。此时,每一条光线相当于单个线程,而 CUDA 的软硬件架构实现了多条线程并行执行,即同时进行多条光线上的相应计算。其大致步骤如下:

步骤一:利用相应函数读入体数据,并将这些在 CPU 中的体数据以三维纹理的形式映射到 GPU 中,同时根据图像的尺寸设定在 GPU 中需要调用的线程块和线程的数目。实验中我们将图像纹理数据的尺寸设为 $512 \times 512 \times 512$,线程块的维度为 16×16 ,即每个线程块中包含 256 个并行执行的线程。

步骤二:计算包围盒。光线投射算法中,视点发出的很多光线并没有穿过物体,是无效的。为了更好的进行三维数据重建,减少不必要的计算量,我们先假设一个包围盒,只有穿过包围盒的光线才有可能通过物体,是应该计算的有效光线。

步骤三:确定光线的起始坐标与方向,并且在内核函数中通过循环实现对一条光线上的体数据的遍历,采样,赋色,光照计算并进行最终效果的累积计算。

步骤四:实现当前光线上采样点颜色的混合,获得最终的图像上每个像素的颜色值,并将所得值写入帧缓存来实现最终图像显示。在混合的过程中,我们既可以采用由前向后的累加,也可采用由后向前的累加,在程序实现过程中我们采用了后者。

3 实验

本文实验所用电脑配置如下:CPU 为双核 Pentium(R) E2180 2.00GHz,内存为 2.00GHz 的 DDR2 型, GPU 为 NVIDIA GeForce 8800 GTS 512,编程环境为 Visual Studio 2005,其中在 CPU (host) 上执行的文件为: application C Program,在 GPU (device) 上执行的文件为: GPU Parallel Kernel

3.1 绘制效果

下图为在采用 CUDA 技术我们所绘制的脑血管体数据时得到的结果,我们不难发现当我们在旋转物体时因为光照的影响,物体会产生明显的明暗对比(见图 3),说明用 CUDA 实现的体绘制光照模型并不比以前的单纯的采用 CPU 或者传统 GPU 加速实现的效果差,而因为并行处理的实现,速度上必然有大幅度的提升。同时由于我们绘制数据的尺寸为 $512 \times 512 \times 512$,这是传统的 GPU 加速难以做到的,因为传统的 GPU

加速因为其纹理尺寸受到限制 ,无法完成如此大规模的体数据绘制。

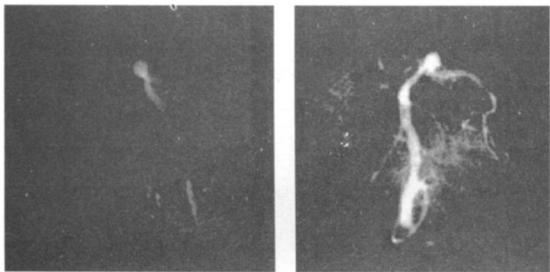


图 3 采用 CUDA 技术最后得到的体绘制 phong 模型，数据尺寸为 512*512*512

3.2 绘制速度的比较

针对不同尺寸的数据集，我们采用传统 GPU 加速算法和基于CUDA的GPU加速分别进行光线投射体绘制，得到时间记录如下表：

表 1 传统 GPU 加速算法耗时与 CUDA 加速耗时对比（帧 / 秒）

数据集名称	数据尺寸	传统 GPU 加速	CUDA 加速
头颅数据 (8bit)	256*256*256	59.0	70.6
头颅数据	512*512*174	33.1	55.6
脑血管数据	512*512*244	25.1	40.5
心脏数据	512*512*276	26.6	43.5
脑血管数据	512*512*512	无法实现	30.3

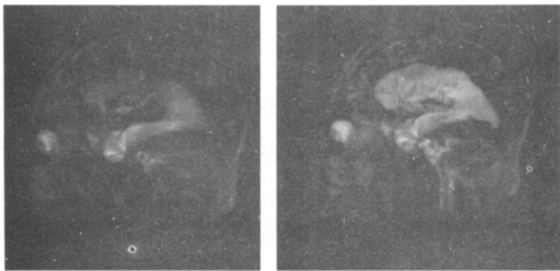


图 4 利用 Cg（左）和利用 CUDA（右）实现的光线投射算法成像对比

通过以上的表格我们可以发现，CUDA 实现的 GPU 加速比传统的 GPU 加速具有更大的优越性，特别是当数据集较大乃至无法使用传统的 GPU 加速来实现时，CUDA 更加显现出强大的优势。并且使用CUDA加速可以让我们观察到更多的图像细节（见图 4）。图 4 中左侧头像为用传统 GPU 加速实现

的光线投射体绘制，中间和右侧的头像为使用 CUDA 技术光线投射所产生的体绘制效果。

结束语

本文主要是针对以前的GPU加速算法在CUDA的环境下进行在加速 ,通过实验的证明 ,在利用CUDA的GPU并行处理后 ,体绘制的效率的确有较大的提高。在以后我们的研究目标将是主要在CUDA 的平台上实现更多的加速算法，并且逐步完善成像的着色效果。

参考文献

[1] 唐泽圣 三维数据场的可视化[M] 北京 清华大学出版社 1999

[2] Daniel Weiskopf, GPU-Based Interactive Visualization Techniques, Springer, 2006, pp.11-32.

[3] 苏超轶 赵明昌 张向文 GPU 加速的八叉树体绘制加速算法计算机应用第 28 卷第 5 期

[4] 储璟骏 杨新 高艳 使用 GPU 编程的光线投射体绘制算法计算机辅助设计与图形学报第 19 卷

[5] Suryakant Patidar, P.J.Narayanan Ray Casting Deformable Models on the GPU

[6] NVIDIA CUDA SDK : http://www.nvidia.com/object/cuda_home.html

[7] I. Boada, I. Navazo, R. Scopigno, Multiresolution volume visualization with a texture-based octree, Visual Comput. 17 (3) (2001) 185-197.

[8] W. Li, A. Kaufman, Texture partitioning and packing for accelerating texture-based volumerendering, Graphics Interface (2003) 81-88

GPU Accelerated Volume Rendering Using the Common Unified Device Architecture (CUDA)

Lei Wu Bin Wang

(Department of Biomedical and information Engineering, Northeastern Univ., Shenyang)

Abstract: In this paper, we present an innovative and optimized version of algorithm based on CUDA, aiming at solving the volume rendering on the basis of traditional method of ray-casting. First, we store the data in the GPU cache in a form of 3D texture, then we use CUDA ' s high level of parallelism to operate on the data, including: generating the texture array to store the value of vertexes, calculating the normal vectors, implementing the coordinate transform, determining the valid rays, calculating the reflection components of each sampled point along the rays, and last using the accumulation function to attain the value for each pixel on the screen. The experiment result shows that by using CUDA we accelerate the rendering process and improve the performance

Key words: Ray-casting, 3D texture, 3D datasets, Phong illumination model, data reconstruction

新闻动态

梅特勒 - 托利多荣获 “ 2009 年高教仪器设备优质供应商 ”

2009 年中国国际教育技术装备展览会暨秋季全国高教仪器设备展示会于 11 月 9 日在广东省东莞市隆重开幕。本次展会安排高校实验室仪器和高教设备专用展区,并在展会期间举办了“ 第一批高教仪器设备优质供应商 ” 评选活动。

评选活动由全国高校实验室工作研究会主办,集教学仪器设备展示和新产品新技术发布于一体,吸引了全国高等教育各级各类院校及众多教学仪器设备厂商的积极参与。

经过参评院校的一致推荐,梅特勒 - 托利多荣获 “ 2009 年高教仪器设备优质供应商 ”,作为推荐供应商在各部直属理工院校、985 院校进行公布及推广,评比结果已安排在《实验技术与管理》、《实验科学与技术》、《实验室科学》、《实验研究与探索》等杂志上刊登,并在 “ 高校实验室工作研究会 ” 网站 (www.culabs.com.cn) 上进行公布。