

文章编号:1007-2780(2013)01-0127-05

基于 GPU 的虚拟内窥镜场景实时绘制算法

朱 兵¹, 付飞蚌²

(1. 中国人民解放军空军航空医学研究所 附属医院, 北京 100089, E-mail: zhubing0001@126.com;
2. 长春理工大学 计算机科学技术学院, 吉林 长春 130022)

摘 要: 为满足影像引导手术(IGS)中高分辨率、海量数据的实时渲染, 提出一种基于 GPU 的虚拟内窥镜场景实时绘制算法。该算法针对虚拟内窥镜渲染数据的特点(管腔数据占总数据比例小, 5%左右), 首先对图像进行自动分割, 得到管腔组织的分割数据; 仅将分割后的数据一次载入图像显存, 利用光线投射算法进行渲染, 并在多 GPU 负载方面做了优化。充分利用 GPU 渲染和并行计算的能力, 实现了海量数据($1\ 024 \times 1\ 024 \times 1\ 024$)的实时渲染。

关 键 词: 虚拟内窥镜; GPU; 光线投影; 负载平衡

中图分类号: TP391 **文献标识码:** A **DOI:** 10.3788/YJYXS20132801.0127

Real-Time Rendering Algorithm of Scene in Virtual Endoscopy Based on GPU

ZHU Bing¹, FU Fei-ran²

(1. Hospital Attached to Aeromedicine Institute of P. L. A., Beijing 100089, China, E-mail: zhubing0001@126.com;
2. School of Computer Science and Technology, Changchun University of
Science and Technology, Changchun 130022, China)

Abstract: This paper proposed a real-time rendering algorithm of scene in virtual endoscopy based on GPU which will meet the real-time rendering of high resolution and mass data in the imaging guided surgery (IGS). According to the characteristics of the virtual endoscopy rendering data that the lumen data is only a very small proportion (around 5%) of the total, the first thing to do is to finish the automatic segmentation of image and to get the segmentation data of lumen tissue. Then the divided data is loaded into the image memory once and make the rendering by ray casting algorithm. In addition, the multi-GPU loading is optimized. Finally, the real-time rendering of mass data is realized through the full use of rendering and parallel computing with GPU, such as $1\ 024 \times 1\ 024 \times 1\ 024$.

Key words: virtual endoscopy; GPU; ray-casting; load balancing

1 引 言

虚拟内窥镜(Virtual Endoscopy, VE)之所以受到广泛的研究与应用是源于其非入侵性和可重复性。19 世纪末, 法国 Laennec Hospital 的研发

小组开发的虚拟食管-支气管内窥镜(Virtual tracheo-Bronchial Endoscopy)^[1-2], 在临床试验与医学诊断中虚拟内窥镜技术已被广泛应用, 主要集中在, 如血管、气管、内耳、结肠等, 那些具有空腔组织结构的器官上。随着医学三维可视化技术的

收稿日期: 2012-11-12; 修订日期: 2012-12-27

作者简介: 朱兵(1963—), 男, 吉林长春人, 大学本科, 副主任技师, 主要研究方向为医疗仪器。

深入发展,用户要求的实时交互处理速度和体绘制速度之间的矛盾越来越严重。当前,可视化医学影像数据集主要有 3 种方式:(1)基于软件加速的方式,通过体数据本身特性优化的体绘制算法得以加速,典型的算法主要有光线终止法、自适应取样法、八叉树分解法、Spattin 和 Shear-sarp;(2)基于硬件加速的方式,通过特定用途的硬件来实现实时的体绘制,典型的硬件主要有 VIZAR 的 TeraRecon 和 VolumePro^[3],然而昂贵的价格限制了图形加速卡的推广和普及;(3)基于 GPU 加速的方式,在引入了通用并行计算平台(Compute Unified Device Architecture,CUDA)与可编程 PC 图形硬件后,基于 PC 图形硬件的编程越来越广泛。

尽管已经提出了多种体绘制算法,但是其本身都存在着一定的不足。因此,本文将提出一种基于 GPU 的虚拟内窥镜场景实时绘制算法,充分利用 GPU 渲染和并行计算的能力,实现高分辨率的、海量数据($1\,024\times1\,024\times1\,024$)的实时渲染。

2 通用平行计算架构

传统的 3D 图形的基本工作都是由 CPU 完成,显卡只是起着帧缓存(FrameBuffer)的作用。1999 年,nVidia 公司推出了具有划时代意义的显示芯片 GeForce256,全球第一款集成了硬件多边形变换和光照(Transformation and Lighting,T&L)等 3D 图形加速功能的可编程图形处理器(Graphics Processing United,GPU),自此,3D 图形芯片踏入了飞速发展期(其更迭速度是 CPU 的三倍多)。

GPU 用来做通用并行计算是在 2003 年^[4],并在该年进入并行计算的主流^[5]。GPU 所特有的并行性、高密度运算和减少了 GPU 与 CPU 的数据通信等优势是采用图形硬件做通用计算的主要意义。

CUDA(Compute Unified Device Architecture,通用并行计算架构)是一个新颖的软硬件相结合架构,其中将 GPU 视为一个并行数据计算的设备,对其运行的计算进行分配与管理。CUDA 的编程核心是,以 Grid、Block、Thread 三层结构,管理和运行最多数万个活动线程。它与传统

的 GPGPU(General Purpose Graphic Process Unit,)所使用的方法截然不同。传统的 GPGPU 进行通用计算时,通过把这些问题转换成图形计算并送到 GPU 中完成^[6],对于 CUDA 则可以直接调用 GPU 的计算资源。鉴于此,本文算法 GPU 计算部分基于 CUDA 实现。

3 光线投射算法

1987 年,Marc Levoy 提出了光线投射算法^[7-9](Ray-Casting),并在科学可视化领域得到迅速的发展。在直接体绘制算法中其中最重要并且最广泛的一类是光线投射。光线投射算法是从成像平面上的每个像素点发出一束光线,将离散采样数据对象的光线,利用传输函数和重构函数获得所有采样点光线数据的颜色值和阻光度值,之后根据不同的光照模型将这些颜色值和阻光度值,进行复合来获得最终像素点的颜色。而对光线投射算法的研究主要在于加快投射速度方面。通过减少成像平面上的采样数来减少所需处理的光线数,可以明显地增加光线投射算法的处理速度。但是,这种做法将降低绘制图像的分辨率。另外一种加速算法则是减少处理每条光线的

时间。例如,快速的跳过不重要的区域,可加快光线的采样混合过程。在从前向后的投射中,我们为了加快光线的处理速度还可以使用光线提前结束

(Early Ray Termination)技术,从而增快了光线投射算法的速度。如何通过并行算法(Parallel Algorithm)来加快光线投射体绘制的速度也是一个研究的热点。其中一类并行算法通过划分体数据,将其分配的数据到每个处理单元,对所有的视角,每个结点仅处理分布在其处理单元上的数据,最后将每个结点产生的子图像复合为最后的绘制图像。另一类并行算法是通过划分图像空间,而

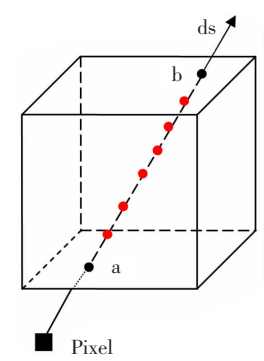


图 1 光线投射算法示意图
Fig. 1 Schematic diagram of ray-casting algorithm

不是数据空间来进行加速绘制过程^[10]。

基于传统 CPU 实现的光线投射算法,在众多体绘制算法中具有灵活体现,绘制质量高等优点,但同时也存在数据运算量大,绘制速度略慢的缺点。因此提出了很多加速算法,但这些加速算法有的是降低了图像质量,有的是加速效果不能满足实时交互的需求。随着显卡的不断发展, GPU 越来越强大,而且 GPU 为显示图像做了更多的优化。在计算图像上已经超越通用 CPU。受到 GPU 强大并行计算能力的驱动,光线投射算法被加以改进,以便能够在 GPU 平台上运行。

Kruger 提出的一种最具代表性的基于 GPU 的多步(Multi-pass)光线投射算法^[10]。但是受当时硬件条件限制,多步光线投射算法实现过程很复杂,需要多次进行绘制体数据,执行效率低。Scharsach 对该方法进行了改进^[11],利用 GPU 的片元着色程序实现循环来执行光线积分,简化了算法的实现。邹华等人提出了层次包围盒与 GPU 实现相结合的光线投射算法^[12],利用层次

包围盒技术实现了光线积分加速,在数据量较小的情况下,加速效果比较明显。袁斌^[13]提出了改进的均匀数据场 GPU 光线投射算法,对数据进行分块渲染,解决了多块数据之间的缝隙问题,但由于分块数据需要动态加载,需要一定的时间开销。

以上的算法都没考虑虚拟内窥镜的特殊情况,即虚拟内窥镜仅要渲染管腔数据,渲染数据量小。

4 算法实现

由于受图形硬件存储空间的限制,当数据量过大不能一次载入显存时,传统基于 GPU 的光线投射算法无法对体数据进行处理。本文提出一种基于视场划分的渲染技术,仅绘制视场内的体素。该算法的基本思想为:将三维体数据映射到世界坐标系,根据视场将提数据划分为可见数据和不可见数据,可见数据载入图形硬件存储空间进行渲染,具体步骤如下:

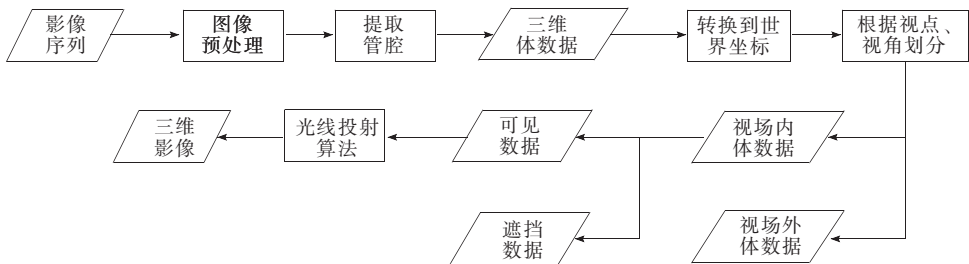


图 2 场景绘制流程图

Fig. 2 Flow chart of scene rendering

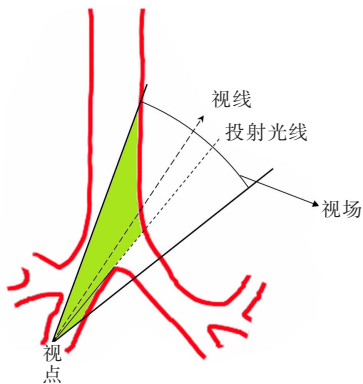


图 3 虚拟管腔划分示意图

Fig. 3 Schematic diagram of virtual lumen divided

(1)载入 DICOM 影像序列,对原始影像预处理,去除图像噪声干扰;

(2)根据人体组织不同管腔密度范围结合 PCNN(Pulse Coupled Neural Network,脉冲耦合神经网络)分割算法^[13],提取管腔像素数据;

(3)将分割提取出的管腔像素数据生成三维体数据并映射到世界坐标系;

(4)根据虚拟镜头在三维场景里的所在视角和位置,可将数据分为两个部分:一部分是视场外数据,由于各种原因,可不进行数据处理;另一方面是视场内数据,需要进行处理。

(5)根据是否有遮挡物遮挡,可将视场内数据进行再次划分,一部分为可见数据,需要进行处理;另一部分由于被遮挡,故不进行处理(虚拟管腔划分如图 3 所示)。

(6)对当前帧可见数据进行体绘制,具体步骤

如下：

(a)生成光线:确定生成的初始光线的起点和方向；

(b)进行场景遍历^[14]:采用类二叉树结构组织场景,利用光线信息在可见场景的空间剖分结构中寻找可能需要进行光线求交的模式表面基元；

(c)模型求交^[15]:进行焦点的探测以及光线与表面基元的求交运算。

(d)渲染与光线衍生^[16]:采用 phone 光照模型进行光照计算,然后将所得数值累加到帧缓存相关像素中,在光线有交时,就利用交点信息生成相应的反射或折射光线就利用交点信息生成相应的反射或折射光线。必要时,可以生成阴影测试

光线。

5 实验结果

本文算法利用 visual C++ 6.0 与 OpenGL 进行实现,其中 GPU 的数据计算部分采用 CUDA 技术实现,在三类数据集上做了测试。数据由吉林大学第一医院提供。所用的机器配置均采用 Intel 酷睿 2 处理器,Duo E6300 的 CPU,机器内存 2 Gbyte,显卡型号为 GeForceFX 5200,显存 128 Mbyte,系统为 Windows XP SP3 系统。利用本文提出的算法实现了管腔图像的实时渲染,其结果如图 4 所示。

实验结果如表 1 所示,它是本文算法与传统体绘制交互速度的比较。

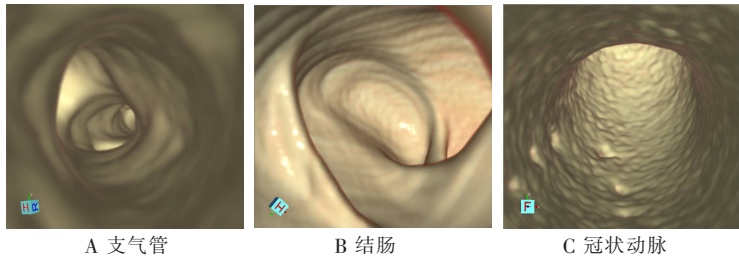


图 4 实时渲染效果图

Fig. 4 Experimental results of the real-time rendering

表 1 本文算法与传统体绘制算法的比较(单位:帧/秒)

Table 1 Comparison of the traditional volume rendering algorithm and proposed algorithm

实验数据	支气管	结肠	冠状动脉
分辨率	1 024×1 024×1 024	1 024×1 024×1 024	1 024×1 024×1 024
传统体绘制算法	9.8	6.4	3.5
本文的算法	36.2	35.7	32.1

从表 1 可以清晰地看出,优化后的算法在绘制速度上有了明显的提升,主要是在在光线生成到渲染片段使用了 GPU 的并行计算。

6 结 论

本文提出的基于 GPU 的虚拟内窥镜场景实时绘制算法,针对虚拟内窥镜渲染数据的特点(管腔数据占总数据比例小,5%左右),首先对图像进行自动分割,得到管腔组织的分割数据;仅将分割后的数据一次载入图像显存,利用光

线投射算法进行渲染,并在多 GPU 负载方面做了优化;充分利用 GPU 渲染和并行计算的能力,实现了海量数据(1 024×1 024×1 024)的实时渲染。此算法在保证不降低三维影像显示精度的前提下,减少光线投射计算中体素的数量,并利用 GPU 的高度并行计算和多数据流处理能力计算机绘制中最复杂的光线生成和渲染片段,预处理时间明显减少,节省了传统算法中 CPU 的占用率。下一步,将优化算法,以适应复杂场景的实时渲染。

参 考 文 献:

- [1] Kagadis G C, Patrino V, Kaloger C P, *et al.* Virtual endoscopy in the diagnosis of an adult double tracheal bronchi case [J]. *European Journal of Radiology*, 2001, 40(9):50-53.
- [2] Razdana A, Patel K, Farin G E, *et al.* Volume visualization of multicolor laser confocal microscope data [J]. *Computers & Graphics*, 2001, 25(5):371-382.
- [3] Kaufman A, Mueller K. Volume visualization and volume graphics [R]. NY: Stony Brook University, 2003.
- [4] Krüger J, Westermann R. Linear algebra operators for GPU implementation of numerical algorithms [J]. *ACM Trans Graph.*, 2003, 22(3):908-916.
- [5] Macedonia M. The GPU enters computing's mainstream [J]. *Computer*, 2003, 36(10):106-108.
- [6] 唐定车,刘任任,谭建龙. 基于统一计算架构的并行串匹配算法 [J]. *计算机应用*, 2009, 29(6):399-401.
- [7] Levoy M. Display of surfaces from volume data [J]. *IEEE Computer Graphics and Applications*, 1988; 8(3): 29-37.
- [8] Levoy M. Efficient ray tracing of volume data [J]. *ACM Transactions on Graphics*, 1990, 9(3):245-261.
- [9] Levoy M. A hybrid ray tracer for rendering polygon and volume data [J]. *IEEE Computer Graphics and Applications*, 1990, 10(2): 33-40.
- [10] Krueger J, Westermann R. Acceleration techniques for GPU-based volume rendering [J]. *Proceeding of IEEE visualization'03*, 2003; 287-292.
- [11] Scharsach H. Advanced GPU ray-casting [J]. *Proceedings of CESC'05*, 2005; 69-76.
- [12] 邹华,高新波,吕新荣. 层次包围盒与 GPU 实现相结合的光线投射算法 [J]. *计算机辅助设计与图形学学报*, 2009, 21(2):172-178.
- [13] 袁斌. 改进的均匀数据场 GPU 光线投射 [J]. *中国图象图形学报*, 2011, 16(7):1270-1275.
- [14] Cox M, Molnar S, Ellsworth D, *et al.* A Sorting classification of parallel rendering [J]. *IEEE Computer Graphics and Algorithms*, 1994; 23-32.
- [15] 常辉,雷小永,戴树岭. 基于动态负载平衡的 sort-first 绘制集群 [J]. *北京航空航天大学学报*, 2010, 36(4):447-450.
- [16] SHI Weili, LI Yanfang, MIAO Yu, *et al.* Research on the key technology of image guided surgery [J]. *Przegląd Elektrotechniczny(Electrical Review)*, 2012, 3(88):29-33.