

基于 CUDA 的体数据可视化工具^①

叶 良, 单桂华, 刘 俊, 迟学斌

(中国科学院计算机网络信息中心 中国科学院研究生院, 北京 100190)

摘 要: GPU 的可编程性和并行计算能力的飞速发展可视化提供了新的解决途径。基于支持 CUDA 的 GPU, 利用光线投射, 实现了一个可以对体数据进行交互式可视化的工具, 包括阻光度融合、等值面绘制、MIP 绘制以及 X 光线投影等多种绘制效果, 并加入了 Phong 光照模型以提高阻光度融合和等值面绘制的图像质量。实验表明, 该工具较好的利用了 GPU 的并行计算能力, 能够绘制出较高质量的图像, 并具有良好的可交互性和可扩展性。

关键词: CUDA; 可视化; 光线投射; Phong 光照模型

CUDA-Based Volume Data Visualization Tool

YE Liang, SHAN Gui-Hua, LIU Jun, CHI Xue-Bin

(Computer Network Information Center, Graduate University of Chinese Academy of Sciences, Beijing 100190, China)

Abstract: The fast development of the Graphics Processing Unit's programmability and parallel computing power provides new solutions for visualization technology. Using ray-casting, a CUDA based interactive volume rendering tool is realized. Several rendering effects have been implemented, including alpha blending, iso-surface rendering, maximum intensity projection and X-ray projection. In order to improve alpha blending and iso-surface rendering effect, a per-pixel based Phong Lighting Model is added in. The experimental results show that the tool can make use of GPU's parallel computing capability and render high quality image. It's interactive and extensible.

Keywords: CUDA; visualization; ray-casting; phong lighting model

可视化(Visualization)技术是指运用计算机图形学和图像处理等技术, 将数据转换化为可以在屏幕上显示的图形或图像, 并进行交互处理的理论、方法和技术。它涉及计算机图形学、图像处理、计算机视觉、计算机辅助设计和人机交互技术等多个领域^[1]。随着计算机技术的不断发展, 可视化的应用范围也得到了极大的扩展, 不仅包括最初的科学计算数据的可视化, 而且包括工程数据和测量数据等的可视化。

随着图形处理芯片(GPU)的可编程性和并行计算能力的飞速发展可视化提供了新的解决途径。2006 年末, NVIDIA 公司推出了具有统一渲染架构的 GeForce 8 系列 GPU, 引入了计算统一设备架构

(CUDA)的概念, 使得 GPU 用于通用计算(GPGPU)的实现更加简单、高效^[2]。

利用离屏渲染技术, 实现了一个基于 CUDA 的单遍光线投射体绘制可视化工具, 包括阻光度融合、等值面绘制、MIP 绘制以及 X 光线投影等多种绘制效果。该工具在某一种绘制的 kernel 内, 先计算投射光线与包围盒的相交情况得到光线参数方程, 接着进行光线投射、光线终止判断以及颜色和阻光度的积累等。为了提高阻光度融合和等值面绘制的图像质量, 还增加了基于 per-pixel 的 Phong 光照模型计算。

1 工具概览及实验平台

工具的基本流程如图 1 所示。

① 基金项目:中国科学院知识创新工程重大项目(KGGX1-YW-13);财政部国家重大科研装备研制项目(ZDYZ2008-2)

收稿时间:2010-04-29;收到修改稿时间:2010-05-20

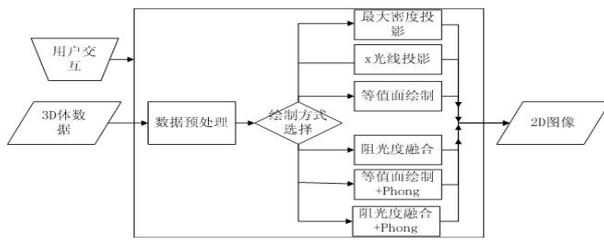


图1 工具基本流程图

实验平台: Windows XP, 开发工具 Visual Studio 2005, CUDA 版本 2.3, CPU 为 Intel E7300, 内存 2G, 显卡为 NVIDIA GeForce 8800 GTS, 显存 640MB, 核心数 96 个, 重建图像分辨率与视口分辨率同为 512*768。

2 基于CUDA的单遍光线投射法

光线投射算法是一种以图像空间为序的经典体绘制算法, 其基本原理是成像平面的每个像素沿着视线方向发出一条射线来穿过体数据, 沿着这条射线进行等间距或不等间距重采样, 求出各重采样点的颜色值和阻光度, 然后按照由前向后或由后向前的方式合成射线上各重采样点的颜色值和阻光度, 即得到该像素颜色^[3]。基于 CUDA 采用后分类由前向后合成的单遍光线投射算法的基本流程如下:

1. 建立离屏渲染所要使用的像素缓冲对象(pbo), 计算投影方向上体数据的包围盒, 并把体数据和传递函数等数据参数信息由 CPU 内存拷至 GPU 全局存储器并进行相应的纹理绑定^[4]。

2. 按照投影空间大小分配适当的线程总数, 将所有线程划分为若干线程块, 每一线程块包含若干线程。建立线程和光线投射之间的映射关系, 其中每一个线程将沿一条光线路径进行合成计算。

3. 沿光线方向循环进行合成计算, 其中每一条光线将在各自指定的线程下并行处理。每次循环时, 首先判断光线是否穿过物体包围盒^[5], 如果没有则直接返回背景颜色; 如果是则从入射点开始, 使用硬件三线插值计算出对体数据纹理的重采样值, 并将其值来查询传递函数纹理, 得到该重采样点的 RGBA 颜色, 再根据光线投射方程进行由前向后的合成计算, 直至光线离开包围盒, 期间还要进行光线终止判断。

4. 将与 pbo 对象绑定的 2D 纹理, 渲染到屏幕指定位置, 得到绘制结果。

3 多种绘制效果

阻光度融合^[6]是在体绘制中应用最广泛的合成方法, 能产生高质量的图像。将沿视线方向重叠的各个重采样点的光学属性根据绘制积分方程进行累积, 每个重采样点根据其位置和光学属性的不同, 对最后的绘制结果都有一定的贡献, 且可通过设置传递函数实现半透明绘制效果。上一章的流程就是以阻光度融合的形式给出的。

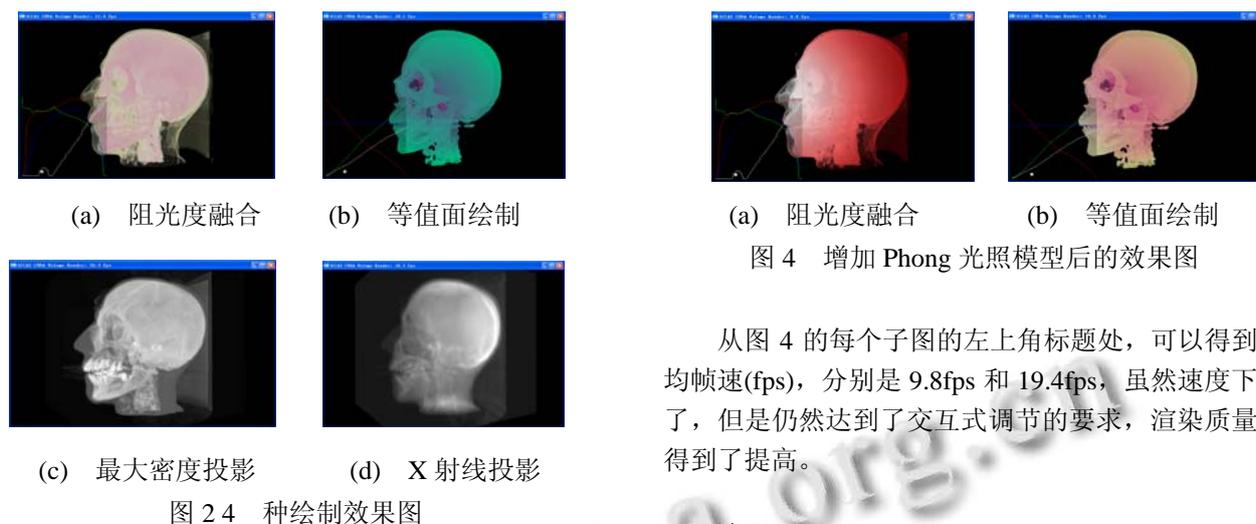
与基于移动立方体方法^[7]的等值面绘制不同, 基于光线投射的等值面绘制原理是视口每个像素投射的光线穿过重建物体, 以一定的步长重采样体数据, 相邻两个重采样点的密度值与阈值相减, 差的符号相反则在这两个重采样点之间存在一个光线与等值面的交点, 插值后可得到交点坐标, 视口上所有交点组成物体在当前阈值下的离散等值面, 之后就可以对等值面进行着色, 如将传递函数作为颜色映射表, 采用位置着色。基于 CUDA 的光线投射算法能快速的抽取任意阈值的等值面, 实现简单, 重建图像质量高。

最大密度投影(MIP)是通过计算沿着穿过被扫描物体的每条射线上所遇到的最大体素密度而产生的。在临床上常用于显示具有相对较高密度的组织结构, 例如注射对比剂后显影的血管、明显强化的软组织肿块以及血管壁的钙化等。基于 CUDA 的光线投射算法实现 MIP, 只需记录视口每个像素所投射的光线穿过重建物体时遇到的最大体素密度值, 经灰度映射赋予该像素灰度值。

X 射线投影是将沿视线方向重叠的一系列重采样点的密度进行加总求平均, 作为该视线对应像素的密度值。这种合成方法产生的图像效果类似于医用的 X 光成像。用基于 CUDA 的光线投射算法实现 X 射线投影, 只需对视口每个像素所投射的光线穿过重建物体时遇到的每个重采样点的密度值累加求平均, 经灰度映射赋予该像素灰度值。

通过交互式调节, 对维度为 256*256*225 的头部数据分别进行 4 种绘制的效果如图 2 所示。

从图 2 的每个子图的左上角的标题处, 可以得到平均帧速(fps), 分别是 21.4fps, 20.1fps, 30.3fps 和 30.3fps, 达到了交互式调节的要求, 而图 2(a)和(b)的左下方是该图像的传递函数, 线条颜色代表了传递函数的 RGBA 颜色分量。



4 Phong光照计算

Phong 光照模型^[8]是一种在计算机图形学中普遍采用的光照模型,其在单一光源照射下的表达式为:

$$I = K_a I_a + K_d I_l \cos \theta + K_s I_l \cos^n \alpha \quad (4-1)$$

其中 I 为光照点处的光强度, K_a 为物体对环境光的反射系数, I_a 为入射光的环境光强度, K_d 为物体的漫反射光系数, I_l 为入射光强度, θ 为入射光与表面法向量的夹角, K_s 为物体的镜面光反射系数, n 为镜面光指数, α 为镜面反射光与视线方向的夹角,如图 3 所示。

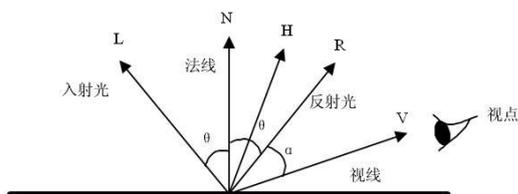
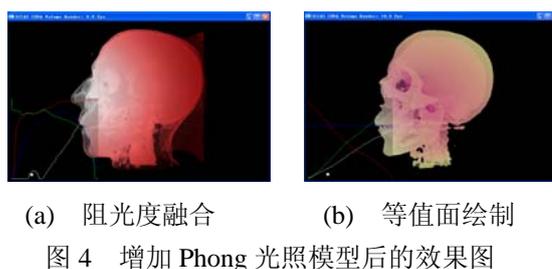


图 3 Phong 光照模型示意图

通过引入实时计算归一化梯度的思想来减少光照计算的误差,利用支持 CUDA 的 GPU 的可编程特性,在需要绘制的空间内部的每个点进行纹理采样,实时计算归一化梯度。然后,基于每个像素(per-pixel)按照光照模型公式计算光照贡献,得到体数据的光照明暗效果。由于充分利用硬件纹理采样的三线型插值操作,同时提高了光照计算的精度,能有效地提高绘制质量。

分别使用和上一章一样的传递函数,在增加了 Phong 光照模型后,经过光照参数调节得到阻光度融合和等值面绘制的效果如图 4 所示。



从图 4 的每个子图的左上角标题处,可以得到平均帧速(fps),分别是 9.8fps 和 19.4fps,虽然速度下降了,但是仍然达到了交互式调节的要求,渲染质量也得到了提高。

5 结语

以医学数据——头部为示例,基于 CUDA 的体数据可视化工具,以单遍光线投射法为基础,实现了阻光度融合、等值面绘制、最大密度投影和 X 射线投影等多种绘制效果,通过加入 per-pixel 的 Phong 光照模型计算,提高了绘制质量,并且可以实现可交互的用户操作,便于用户分析数据特征,也容易在此工具的基础上改进或扩充新的功能。

参考文献

- 1 Post FH, Gregory M. Nielson and Georges Pierre Bonneau. Data Visualization: The State of the Art. Massachusetts: Kluwer, 2002:1-3,93-94.
- 2 NVIDIA Corporation. NVIDIA CUDA Compute Unified Device Architecture: Programming Guide Version 2.3. Santa Clara, California, 2009. 1-14.
- 3 Levoy M. Display of surfaces from volume data. IEEE Comp. Graph. & Appl., 1988,8(5):29-37.
- 4 叶良,单桂华,迟学斌.基于 CUDA 加速的光线投射法研究.北京:北京交通大学出版社,2010.235-240.
- 5 Owen GS. Ray-Box Intersection. April 01, 1998. <http://www.siggraph.org/education/materials/HyperGraph/raytrace/rtinter3.htm>. October 21, 2009.
- 6 Kaufman A, Mueller K. Visualization Handbook, Chapter 7 Overview of Volume Rendering. Massachusetts: Academic Press, 2005. 127-174.
- 7 Lorensen WE, Cline HE. Marching cubes: a high resolution 3D surface construction algorithm. Computer Graphics, 1987, 21(4):163-169.
- 8 Phong B. Illumination for computer generated pictures. Communications of the ACM, 1975,18(6):311-317.