基于 GPU 与 Monte-Carlo 的体绘制光线投射算法的研究

张春凯，杨 猛，刘金刚
（首都师范大学 计算机科学联合研究院，北京 100040）

摘 要：体绘制过程中等距离采样在显示效果不理想的情况下，每减少一个采样步长会增加大量采样点，大大增加了体绘制过程中的计算负担。针对这个问题，提出了一种基于 Monte-Carlo 积分方法的光线投射实现的实时体绘制算法，采用 Monte-Carlo 积分方法解决了光照明方程中的积分问题。实验结果表明，在显示效果几乎一样的前提下，采用本文的方法绘制效率提高了十多帧。

关键词：体绘制；光线投射；蒙特卡洛
中图分类号：TP39 文献标识码：A 文章编号：1674-7720(2011)18-0042-04

The research of volume raycasting based on CPU and Monte-Carlo

Zhang Chunkai, Yang Meng, Liu Jin'gang
(Join Faculty of Computer Scientific Research, Capital Normal University, Beijing 100040, China)

Abstract: In the volume rendering process in which the equal step sampling is used when the display is not ideal, a large number of samples will increase when a sample step is reduced. It increases the data calculation in the volume rendering process greatly. For this problem, this paper proposes a method based on Monte-Carlo integration to achieve real time ray casting volume rendering. The paper uses the Monte-Carlo integration method to solve the integral equation in the light propagation. The experimental results show that when the premise of visual effect is almost the same, the method used in this paper would improve the efficiency in drawing more than ten frames.

Key words: volume rendering; ray casting; Monte-Carlo

自 20 世纪 80 年代科学计算可视化(Visualization in Scientific Computing)的概念被提出后，三维体数据的可视化技术便开始成为一个独立的研究领域，并最终形成了体绘制技术体系[1]。科学计算可视化的基本含义是运用计算机图形学的原理和方法，将科学与工程计算产生的大规模数据转换为图形、图像，以直观的形式表现其物理属性或统计属性。直接体绘制技术是科学可视化的重要研究内容，目前在许多领域得到了广泛应用。传统的直接体绘制有着绘制速度慢、交互性差等缺点。

体绘制技术是依据三维数据，将所有的体细节同时展现在二维图像的技术，利用体绘制技术，可以在一幅图像中显示更多物质的综合分布情况，并且可以通过不透明度的控制，反映等值面的情况[2]。由于体数据通常具有信息量大、绘制复杂度高的特点，很难满足人们实时显示跟交互的需求，因此，体绘制必须与相应的加速技术和策略相结合。现有的体绘制加速技术主要分为空体素剔除、提前不透明度截止、硬件加速三类，其中前两类属于软件加速，利用对体数据的处理、体数据的渲染流程进行优化等方法来达到加速的目的，但是这种加速方法加速有限，很难达到实时交互。第三类硬件加速又可分为专用图形硬件加速与通用图形处理器加速，但是专用图形硬件成本高，而通用图形处理器比较普及，尤其是可编程 GPU(Graphics Processing Uint)的出现，为实现实时体绘制技术提供了强大硬件支持。

基于 GPU 的加速技术首先由 Cullip 和 Neumann[3]提出，Cabral 等人对这项技术作了改进并验证了体绘制通过硬件加速的可行性[4]，KRUGER J 等人将提前不透明度截止、空体素剔除技术等应用到基于 GPU 的体绘制中，进一步提高了体绘制的速度。

本文采用了 Monte-Carlo 方法计算光照方程。本文描述了基于 GPU 的光线投射算法的总体流程，详细描述了如何将 Monte-Carlo 积分法运用于全局光照模型中，

实验表明，采用随机采样的 Monte-Carlo 方法比等步长采样的黎曼方法具有更好的可交互性。

1 光线投射算法

1.1 光线投射算法

光线投射算法是一种基于图像序列的直接体绘制算法。其原理是从图像的每一个像素沿固定方向发射一条射线，沿着该射线对离散数据进行等距离采样。通常使用三线性插值作为重建滤波器。简言之，每个重采样点的值通过一个映射表产生一个 RGBA 四元组，该四元组封装了该点的发射与吸收参数等光学属性[5]。然后体渲染积分器会通过从前向后或者从后向前的方式将颜色与透明度混合来计算该点的像素颜色值，通常的方式是从前向后混合，原理图如图 1 所示。

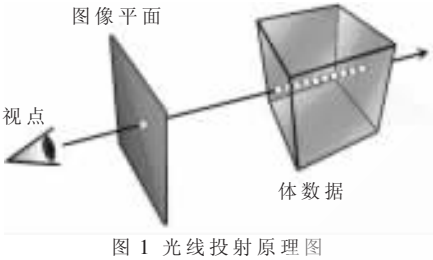


图 1 光线投射原理图

早期的光线投射算法完全基于软件实现，绘制效率受到较大的限制。随着可编程图形硬件的发展，越来越多的光线投射算法开始基于硬件实现，达到了比纯软件算法高出一个数量级的计算效率。

1.2 基于 GPU 的光线投射算法

基于 GPU 的光线投射技术是将整个渲染体(volume)存储在一个简单的 3D 纹理中，然后调用片段程序向渲染体中投射光线。在渲染体中的每一个像素/片段都对应了一条射线，这条射线的参数方程可用式(1)表示：

$$X(t,x,y)=c+t\vec{d}(x,y) \tag{1}$$

其中，规范化的方向向量 $\vec{d}(x,y)$ 可以由像素的屏幕坐标 (x,y) 与相机的坐标 c 计算得到或者通过光栅化得到[6]。在光线投射片段程序运行之前，每一个片段(像素)射线进入与射出渲染体的位置 $[t_{start}(x,y),t_{exit}(x,y)]$ 在光线投射程序执行之前会事先计算。简单地说，$t_{start}(x,y)$ 表示的是渲染体的前表面离摄像机的距离或者渲染体包围盒的前表面经过光栅化得到的值，而 $t_{exit}(x,y)$ 表示的是渲染体的后表面距离射线机的距离或者渲染体包围盒的后表面经过光栅化后得到的值。

图 2 说明了通过光栅化对光线投射的入射点、射出点、方向进行初始化。从图 3 可知，光线的起点是由体包围盒的前表面决定，终点是由体包围盒的后表面决定的。光线在这个空间里进行采样，通常以一个恒定的采样速率进行。现在的 GPU，一个简单的渲染通路和光线投射的片段着色程序，通过由前向后的顺序渲染渲染体，就可以生成深度图片(如图 1 所示)。基于 GPU 的光线投射算法是在此基础上应用了空体素剔除与提前不透明度等加速算法，其算法流程如图 4 所示。

图 2 光线投射的初始化，由后表面(中)减去前表面(左)得到射线的方向与长度(右)

图 3 光线投射过程中，在 (f_0-f_4) 与 (l_0-l_4) 之间通过光栅化进行采样

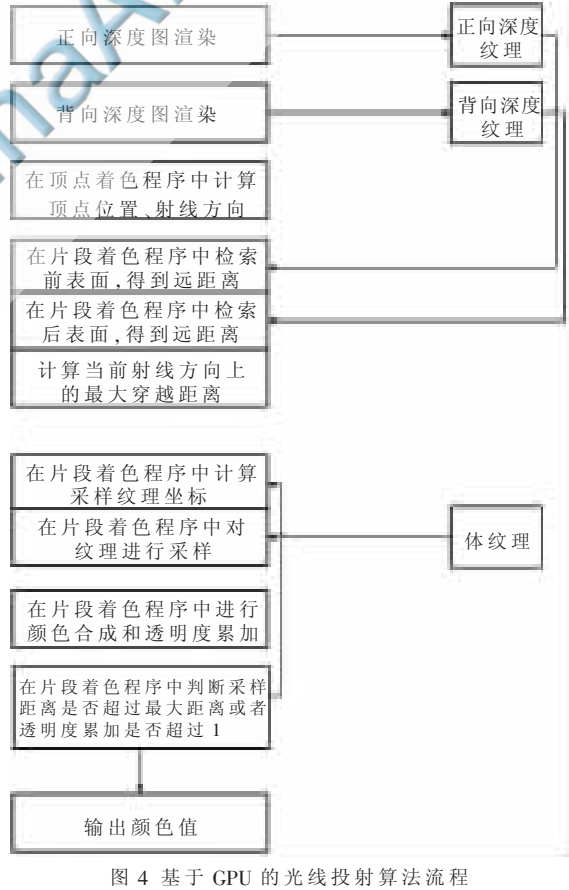


图 4 基于 GPU 的光线投射算法流程

2 在体绘制中应用 Monte-Carlo 方法

2.1 散射效果

散射是使光改变其直线传播的一个物理过程，散射现象包括光的折射、反射、透射等现象。简单的散射现象使用 Phong 光照模型来模拟，举例来说，如果光线照射在一个球形表面上，散射光依据球的表面法线均匀地分布在整个球形表面之上，而镜面光则作用在以几何反射光线为中心的一个扇形区域。更复杂的模型材质效果需要利用双向反射分布函数 BRDF(Bidirectional Reflectance Distribution Function)来描述。

在某一表面上一点 x 的双向反射分布函数(BRDF)$f_r(x,\lambda,w_i,w_o)$说明了光线在该点的反射情况。该函数的参数是：入射光的方向 w_i、反射光的方向 w_o、光线的波长 λ，返回值是反射光强占入射光强的百分比。在很多应用程序中，特别是实时的程序中，局部的光照明模型利用点光源组合而成，也就是说光源由一个或者几个独立的光源组成，光线的方向可能有一个或者几个方向。一个光源应用双向反射分布函数得到的渲染公式如下：

$$L(x,w_o)=f_r(x,w_i,w_o)\cos\theta_i\frac{L_i}{\pi r^2} \tag{2}$$

其中，$L(x,w_o)$是从表面的一点沿着 w_o 方向发出的光强，L_i 是光源发射出的光强，r 是这点与光源之间的距离，θ 是入射光 w_i 与此点法线之间的夹角。在现实中，入射光可能来自于正向球面 Ω^+的所有方向，此时，观察到的光强是此球面上所有的入射光 w_i 积分得到的，公式如下：

$$L(x,w_o)=\int_{\Omega^+} f_r(x,w_i,w_o)\cos\theta_i L(x,w_i)\mathrm{d}w_i \tag{3}$$

前面一直讨论的是位于表面的散射效果，它可以用双向反射分布函数来描述。在半透明体或者粒子系统里，散射可以认为是发生在物体内部的每一个点，在这种情况下双向反射分布函数 f_r 将被相位函数 h 取代，入射光强通过对整个球面进行积分得到，公式如下：

$$L_o(x,w_o)=\int_{\Omega} h(x,w_i,w_o)L_i(x,w_i)\mathrm{d}w_i \tag{4}$$

相位函数 h 描述了相关介质的散射特性，它可以直接操作光照的辐射度值，最常用的相位函数模型有 Henyey-Greenstein[7]、Schlick、Mie[8]和 Rayleigh[9]。

2.2 Monte-Carlo 积分

根据体数据生成逼真的图片，则需要使用恰当的方法计算积分公式，由式(4)可以计算在给定点 x 的特定方向 w_o 的光照强度。此积分只能通过适当的数值积分而不能通过直接计算得出。数值积分用到的最多的方法是黎曼积分方法。对于一个简单的一次积分 $I=\int_a^b f(x)\mathrm{d}x$，通过黎曼积分解此积分是将定义域划分为 n 个相等的区间，然后通过计算相应矩形区域的面积的和得到：

$$I=\int_a^b f(x)\mathrm{d}x\approx\sum_{i=0}^{n-1} f(x_i)\frac{b-a}{n} \tag{5}$$

其中，$x_i=i\cdot\frac{b-a}{n}$。这种方法的误差取决于区间划分的精细程度，如果区间被划分为趋近于无穷块，那么通过这种方法得到的结果误差就趋近于 0。

如果修改黎曼积分，使采样区间(采样点)在随机的位置，假定采样点服从均匀分布，在区间上，则估算值其中在区间服从均匀分布，x 在区间[a,b]上，则估算值 $I'=\sum_{i=0}^{n-1} f(x)\frac{b-a}{n}$，其中 x 在[a,b]区间服从均匀分布 $p(x)$，$p(x)=\frac{1}{b-a}$。

通过以下证明可以得出 I' 的期望与 I 相同：

$$\begin{aligned}E[I']&=E[\sum_{i=0}^{n-1} f(x_i)\frac{b-a}{n}]\\&=\frac{b-a}{n}\sum_{i=0}^{n-1}E[f(x_i)]\\&=\frac{b-a}{n}\sum_{i=0}^{n-1}(\int_a^b f(x_i)p(x_i)\mathrm{d}x)\\&=\frac{b-a}{n}\sum_{i=0}^{n-1}(\int_a^b f(x_i)\frac{1}{b-a}\mathrm{d}x)\\&=\frac{1}{n}\sum_{0}^{n-1}(\int_a^b f(x_i)\mathrm{d}x)\\&=\int_a^b f(x)\mathrm{d}x\\&=I\end{aligned}$$

由此说明 I' 的估算值与积分 I 的值相同，I' 被称为 Monte-Carlo 积分。本文在原有的基于 GPU 的光线透射算法基础上运用了 Monte-Carlo 方法计算式(4)，得到结果，从而使 GPU 在计算过程中采样点减少，速度加快，可交互性增强。

在计算机图形学中，使用黎曼积分可能会存在以下两个缺点：

(1)如果 $f(x)$频率特别高，等距离采样不可避免地会造成走样，即使采样点的数目增加特别多，走样现象也还是会特别严重；

(2)黎曼积分依赖于数据的结构。举例而言，如果对一个三维数据进行采样，可能会采集 10×10×10=1 000 个采样点，若此时发现得到的结果产生的图像错误非常大，接下来可能会采用 11×10×10=1 100 个采样点，即在一个方向多采集一个点，就会导致整个采集样本增加 100 个额外的采样点，造成数据量激增。

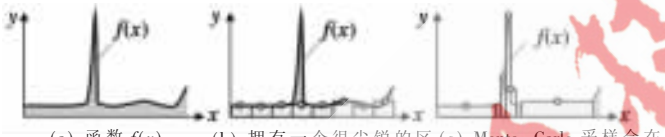
采用 Monte-Carlo 方法可有效地避免以上两点，针对问题(1)，采用 Monte-Carlo 方法下的随机采样，可能会产生一定的扰动，但是视觉效果比走样更加容易让人接受，因为人们的眼睛对于走样要比扰动敏感的多。对于问题(2)，随机采样更加容易避免，不受数据格式的影响，随机采样可以每次只增加一个采样点，直到达到要求的精度为止。

普通的 Monte-Carlo 积分公式为：

$$I'=\frac{1}{n}\sum_{i=0}^{n-1}\frac{f(x)}{p(x)} \tag{6}$$

在本文算法中，$f(x)$值变化剧烈区域需要将区间划分为更多块(取更多的采样点)，值变化平缓的区域则取少量的采样点，如图 5(c)所示。在式(4) $L_o(x,w_o)=\int_\Omega h(x,w_i,w_o)L_i(x,w_i)\mathrm{d}w_i$ 其中，$h(x,w_i,w_o)$的信息是可以计算的，实验过程中就可以在相位函数 h 值变换剧烈区域采取更多的采样点，而在 h 值变化平缓的区域采取相对较少的采样点。如果知道光照方程 $L_i(X,w_i)$的相关分布，则也可以采取相应的方法。这种方法叫做重点采样，可以加快 Monte-Carlo 积分的收敛。实验过程中如果采取黎曼积分，进行等步长采样，就很有可能会错过 $f(x)$值变换剧烈的区域，如图 5(b)所示，在计算积分的过程中使用重点采样的 Monte-Carlo 方法可以避免这个问题，因为 Monte-Carlo 的采样步长 w_i 根据概率分布函数来计算：

$$w_i=\frac{1}{n\cdot p(x_i)} \tag{7}$$



(a) 函数 $f(x)$　(b) 拥有一个很尖锐的区域，等距离采样很容易错过尖锐区域　(c) Monte-Carlo 采样会在尖锐区域取多个采样点

图 5 采样图

3 实验结果分析

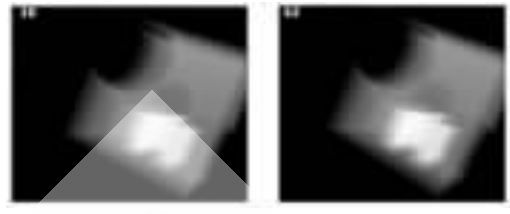
本文实验所用的硬件配置是Intel Core DUO 2.2 GHz CPU、2 GB 内存、NVDIA GeForce GT240M 显卡，操作系统为 Windows7，采用 OpenGL 与 CG 着色语言。

本文实验 1 所用的数据为头部 CT 数据图像，体数据分辨率大小为 256×256×225。使用 GPU 与 Monte-Carlo 方法进行体绘制，得到的效果如图 6(a)所示，绘制帧率为每秒 68 S/s。使用基于 GPU 与等步长采样进行体绘制，得到的效果图如图 6(b)所示，绘制帧率为 50 S/s。显示效果几乎相同的前提下，前者比后者每秒快 18 帧，大大提高了体绘制的可交互性。

实验 2 采用的体数据分辨率为 256×256×256，使用 Monte-Carlo 方法进行体绘制，得到的效果如图7(a)所示，帧率为 60 S/s。使用基于 GPU 与等步长采样进行体绘制，得到的效果图如图 7(b)所示，帧率为 38 S/s，前者比后者每秒多 22 帧。对比可得，本文方法提高了体绘制的可交互性。

(a) 基于 Monte-Carlo 的光线投射体绘制，帧率为 68 S/s　(b) 采取等步长采样的光线投射体绘制，帧率为 50 S/s

图 6 实验 1 结果



(a) 采取等步长采样的光线投射体绘制，帧率为 38 S/s　(b)基于 Monte-Carlo 的光线投射体绘制，帧率为 60 S/s

图 7 实验 2 结果

由实验结果可以看出，基于 GPU 与 Monte-Carlo 的光线投射体绘制有很大的加速效果。

本文描述了基于 GPU 的光线投射体绘制机制，用 Monte-Carlo 积分替代等步长采样来计算光照方程，避免了体数据结构的依赖性，有效消除了失真走样。实验结果表明，本文的算法在基本保持原有视觉效果的前提下大幅提高了渲染效率。

参考文献

[1] HEARN D, BAKER M. 计算机图形学[M].北京：电子工业出版社，2002.

[2] 康玉之.GPU 编程与 CG 语言之阳春白雪下里巴人[M].[出版地不详]：[出版者不详].2009[2010.20]//www.docin.com/P-34717438.html.

[3] CULLIP T, NEUMANN U. Accelerating volume reconstruction with 3D texture hardware[R].Tech.Rep.TR93-027, University of North Carolina at Chapel Hill, 1993.

[4] CABRAL B, CAM N, FORAN J. Accelerated bolume rendering and tomographic reconstructions using texture mapping hardware [C]. In Proceedings ACM Symposium on Volume Visualization,1994:91-98.

[5] LOEVY M. Display of surfaces from volume data[J]. IEEE Computer Graphics and Applications,1988,8(3):29-37.

[6] KRÜGERJ, WESTERMANN R. Accleration techniques for GPU-Based volume rendering[C]. In Proceeding IEEE Visual ization 2003, 2003.

[7] HENYEY L, GREENSTEIN J. Diffuse radiation in the galaxy[J]. Astrophysical Journal,1941,93.

[8] ENGEL K, HADWIGER M,KNISS J, et al. Real-time volume graphics[M]. AK Peters, 2006.

[9] PHARR M, HUMPHRIES G. Physically based rendering [M]. Morgan Kauffman,2004.

(收稿日期：2011-01-17)

作者简介：

张春凯，男，1988 年生，主要研究方向：计算机应用技术。