

一种高效的光线投射体绘制算法

方 军,房晓阳

(南京理工大学 计算机科学与工程学院,江苏 南京 210094)

摘 要:作为体绘制中的一个经典绘制算法,光线投射算法理论简单同时能产生高质量的图像,被广泛应用于医学图像可视化领域。但在绘制过程中有大量的投射光线和体素的重采样,导致绘制速度较为缓慢。为提高绘制的速度,文中提出一种高效的光线投射体绘制算法,通过引入碰撞检测技术减少投射光线的数目,避免冗余光线的采样计算,同时采用光线跳跃方法在碰撞检测包围盒内跳过对空体素的重采样,加快了光线合成的过程。实验结果表明,改进后的算法不仅能保证所需要的图像质量,还能大幅度地减少采样计算的时间,高效地提高绘制速度。

关键词:光线投射;体绘制;光线跳跃;碰撞检测;包围盒

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2014)08-0067-04

doi:10.3969/j.issn.1673-629X.2014.08.016

An Efficient Ray Casting Volume Rendering Algorithm

FANG Jun, FANG Xiao-yang

(School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China)

Abstract: As a classical rendering algorithm of volume rendering, ray casting algorithm has a simple theory and can produce high quality images at the same time, which is widely used in the field of medical image visualization. But, it produces a large number of projecting rays and voxels resampling in the process of rendering, leading to the slow rendering speed. In order to improve the speed of rendering, present an efficient ray casting volume algorithm, the algorithm introduces the technology of collision detection to reduce the number of projecting rays, avoiding redundant ray sample calculations, while using ray leaping method to skip empty voxel resampling in the collision detection bounding box, speeding up the process of synthesis of ray. The experimental results show that, the improved algorithm not only guarantees the image quality required, but also significantly reduces the sampling calculation time, effectively improving the rendering speed.

Key words: ray casting; volume rendering; ray leaping; collision detection; bounding box

0 引 言

近年来,医学图像可视化技术在医学图像处理中已经成为一个热门的话题。利用可视化技术可以获得人体组织和器官的内部信息,将其信息构造成三维图像,可以更加方便地观察其内部的真实构造情况,成为医生进行诊断和治疗疾病的重要辅助手段。医学三维数据的可视化方法分为两类:面绘制和体绘制^[1]。体绘制包括光线投射算法、足迹表算法、错切变形算法、抛雪球算法和纹理映射算法。其中光线投射算法是一个经典的图像空间扫描的体绘制方法,绘制图像质量最高,应用最为广泛。但由于该算法计算量大,绘制时间比较长,达不到实时交互的要求,因此限制了它的应

用。为提高绘制的速度,许多研究者提出了关于光线投射算法的加速技术。利用光线相关性,Lakare 和 Kaufman^[2]提出了一种空间跳跃加速体绘制技术;樊鹏等人^[3]提出改进求交的自适应光线投射算法,确定自适应采样步长;文献[4]提出片段融合的光线投射算法,减少了插值计算量和对无效平面的求交;文献[5-7]提出 GPU 加速的光线投射算法。这些算法虽然有一定的加速,但还没有达到实时绘制的要求。目前研究的热点是如何用更少的投射光线和采样点数目进行光线投射。文中结合碰撞检测包围盒技术和光线跳跃方法的优缺点提出了一种高效的算法,在不影响图像质量的情况下提高了绘制速度。

收稿日期:2013-10-17

修回日期:2014-01-20

网络出版时间:2014-05-21

基金项目:国家重大科学仪器设备开发专项(2012YQ05025004);国家自然科学基金资助项目(61071146)

作者简介:方 军(1988-),男,安徽阜阳人,硕士,研究方向为图像处理、三维可视化。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20140524.2150.034.html>

1 传统光线投射算法

M. Levoy^[8]提出的光线投射算法就是一种图像空间扫描的实现体绘制的离散方法,其基本原理是:从图像空间屏幕上的每一个像素点出发,根据设定的视线方向发出一条射线,这条射线穿过三维数据场,然后沿着这条射线选择若干个等距的采样点,并由距离此采样点最近的 8 个数据点的颜色值和不透明度值作三次线性插值,求出该采样点的颜色值及不透明度值,最后将这条射线上各采样点的颜色值及不透明度值由前向后或由后向前加以合成,即可得到发出该射线的像素点处的颜色值,从而得到屏幕上最终的图像。原理如图 1 所示。

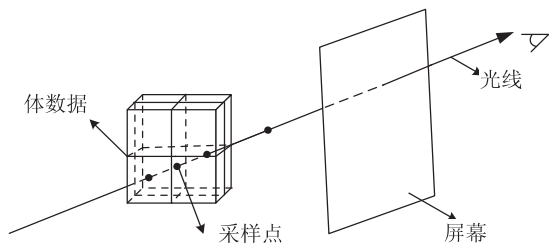


图 1 光线投射基本原理

为了更加清晰地显示体数据的内部结构和层次关系,产生真实、高质量的图像,光线投射算法使用了数据场内所有体素对像素点的贡献,因此就需要屏幕上的所有像素点发出射线和在每一条射线上选择更多的采样点。而许多射线和空体素对于图像合成没有任何贡献,但也需要进行采样处理,导致采样计算量相当大,绘制速度非常缓慢,降低了光线投射的实时交互。

2 改进的光线投射算法

2.1 碰撞检测算法

在虚拟场景中有许多对象,如飞机、汽车、山川、高楼等。碰撞检测^[9-10]就是检测这些对象之间是否会发生碰撞,如汽车撞上了高楼等。碰撞检测的原理就是用包围对象的简单几何体代替复杂的对象,当判断两个对象是否相交时,只需要检测两个几何体的求交测试问题,简化了问题的求解方式。碰撞检测算法包括包围盒法、距离跟踪法和空间分解法等。

包围盒碰撞检测算法是比较容易理解的,不对场景中的对象进行检测,而只对包围其自身的几何体进行求交测试。最常见的包围盒算法有 AABB 包围盒 (Axis-Aligned Bounding Box)^[11]、包围球 (Sphere)^[12]、方向包围盒 OBB (Oriented Bounding Box)^[13] 及固定方向凸包 FDH (Fixed Directions Hulls 或 k -DOP)。文中利用方向包围盒碰撞检测算法优点进行研究工作。

方向包围盒 OBB 是一个比较常用的包围盒类型。一个给定对象的 OBB 被定义为包含该对象且相对于

坐标轴方向任意的最小长方体。OBB 最大特点是它的方向任意性,这使得它可以根据被包围对象的形状特点尽可能紧密地包围对象,能比较显著地减少包围盒的数目。在可视化的三维体数据中,只含有少量的组织和器官,在体数据外存在大量的无效空体素,使用包围盒把那些要采样的体素包围起来,而把那些体数据外围的空体素隔在包围盒的外面,只对包围盒内的采样点进行插值和光线合成,这样就能够减少无效空体素的重采样。根据体数据的大小建立包围盒,在三维空间中设定 OBB 的中心为 C ,沿物体坐标轴方向的三个相互正交的方向向量为 v_1, v_2, v_3 ,在这三个方向上的边长为 r_1, r_2, r_3 ,则 OBB 包围盒所确定的区域可以表示为 $R = \{C + ar_1v_1 + br_2v_2 + cr_3v_3 \mid a, b, c \in (-1, 1)\}$,整个包围盒空间坐标点的值都落在区域 R 内,有效的体素都在包围盒内,把那些无效的空体素隔在包围盒外。算法首先判断射线是否与包围盒相交,若相交则在包围盒内进行光线采样计算;若不相交,则进行下一条光线的合成。这样可以减少投射光线的数目,提高绘制速度。

2.2 光线跳跃算法

光线跳跃^[14]的基本思想:在体数据内有许多空体素,如果每条光线合成都计算所有的体素,那将耗费很大的采样计算时间,所以在光线跳跃算法中会跳过那些不需要进行重采样的空体素,只对那些有效的体素进行重采样。光线追踪不需要每次都一步一步向前追踪,而是预先设置追踪的向前步长大小和向后步长大小。当遇到一个透明体素时,继续按照向前步长大小向前跳跃,当追踪时遇到一个非透明的体素时,先进行采样处理,然后光线向后一步一步进行体素的三次线性插值重采样,当采样数目等于向后步长小时结束向后移动,然后返回到先前非透明的体素继续向前跳跃追踪。直到射线出了数据场或光线合成的不透明度值达到要求的阈值才结束该条射线的合成,继续下一条光线的合成。

2.3 改进算法的原理

文中改进算法的主要思想就是在光线投射的过程中使用更少的投射光线和尽可能地只对不透明体素进行重采样,以减少采样计算的时间,提高绘制速度。其基本原理:在进行光线投射算法之前,首先做一些预处理,即选择合适的 OBB 包围盒包裹要进行可视化的体数据,设置适当的光线跳跃向前步长大小和向后步长大小。然后才从屏幕上的每个像素点发出射线,当射线进入三维数据场时,不立即进行采样处理工作,而是首先判断射线是否与包裹体数据的包围盒相交(即射线能够穿过包围盒),若不相交,则对于这条射线上的所有体素不进行采样处理,因为它们不对图像有任何

贡献;若相交,则求出该条射线与包围盒的两个交点(入盒点与出盒点),然后在包围盒内进行光线跳跃采样计算,直到射线出了包围盒或光线合成的不透明度值达到要求的阈值才结束该条射线的合成,继续下一条光线的合成。原理如图2所示。

向前步长大小=4个体素
向后步长大小=8个体素

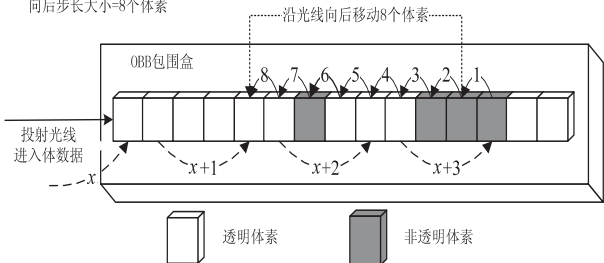


图2 改进的算法原理图

改进的算法中利用包围盒技术能够减少投射光线的数目,缺点是虽然减少了投射光线的数目,但是包围盒内的光线合成还是用传统的光线投射算法,对于每一个体素都要进行三次线性差值,尤其是那些空体素也进行重采样,增加了采样计算的时间,因此如何在包围盒内跳过空体素而只对不透明的体素进行重采样成为研究的重点。而光线跳跃算法的优点是在投射光线上进行三次线性插值时会跳过空体素而只对不透明的体素进行重采样,缺点是还需要对屏幕射出的所有光线进行采样处理,即使有些光线没有穿过体数据,对于图像的生成没有一点贡献,但也会做一些无用的采样计算,在一定程度上减缓了绘制的速度。

通过对上面两种算法的分析,可以看到它们的优缺点刚好能够进行互补,改进的算法不仅能减少投射光线的数目,还能够大幅减少体素采样的数量,提高绘制的速度,同时不降低图像质量。

2.4 改进算法的伪代码

改进算法的伪代码如下所示。

```
Set all pixels to BACKGROUND
Set forward_step_size
Set back_step_size
Set threshold
for each pixel (i,j) in the screen do
    [u,v,w] := ray_enter_box(i,j);
    [x,y,z] := ray_exit_box(i,j);
    Initialize_alg([u,v,w],[x,y,z]);
    len := Length([u,v,w],[x,y,z]);
    while( len = len - forward_step_size )
        if( Opaque( Volume[u,v,w] ) ) then
            if( CompleOpacity( Volume[u,v,w] ) > threshold )
                draw_pixel(i,j, Volume[u,v,w]);
            exit_while;
        [d,e,f] = [u,v,w];
        for( stepback = 1 to back_step_size )
```

```
[a,b,c] := previous_voxel_ray([u,v,w], stepback);
if( Opaque( Volume[a,b,c] ) ) then
    if( CompleOpacity( Volume[a,b,c] ) > threshold )
        draw_pixel(i,j, Volume[a,b,c]);
    exit_while;
[d,e,f] = [a,b,c];
end_if
end_if
end_for
end_if
end_if
draw_pixel(i,j, Volume[d,e,f]);
[u,v,w] := next_voxel_ray(forward_step_size);
end_while
end_for
```

由上述代码可以看出,函数 ray_enter_box, ray_exit_box 为进出包围盒时返回边界点的位置,函数 Initialize_alg 判断像素点发出的射线是否在包围盒内,若在进行重采样,若不在进行下一个像素点的光线采样。函数 CompleOpacity 计算此时进行合成的采样点累加的不透明值,判断其与预先给定的阈值进行比较,若大于则画像素点,停止采样,继续下一个像素点的光线合成。函数 previous_voxel_ray 向后进行遍历采样,函数 next_voxel_ray 按向前步长大小进行跳跃采样。

3 实验结果

为了验证文中提出的改进算法,设置了图像质量和绘制速度的比较实验,并且对实验结果进行了比较和分析。实验对象中的数据来自 <http://www9.informatik.uni-erlangen.de/External/vollib/>, 体数据为 Chapel Hill CT Head, 数据大小为 $256 * 256 * 113$, 数据格式是 .raw。实验环境的计算机配置为

1) 硬件配置:四核 Intel(R) Xeon(R) CPU W3520 @ 2.67 GHz/2.66 GHz, 4 GB 内存, NVIDIA Quadro FX 580 显示卡, 500 G 硬盘空间;

2) 软件配置: Win7 旗舰版操作系统, VS2010。

实验结果如图3和图4所示。

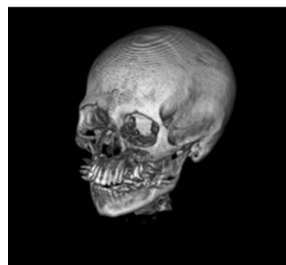


图3 传统光线投射算法

图3是用传统光线投射算法得到的三维图像效果,图4是用文中改进的光线投射算法得到的三维图

像效果。按照图 3 和图 4 所示,两种算法都能产生所需求的高质量图像,两者并没有明显的差别,都有很好的图像效果,说明改进的算法保证了图像质量。表 1 为文中改进算法与传统算法、基于包围盒算法和基于光线跳跃算法在绘制时间和旋转时间上的比较。

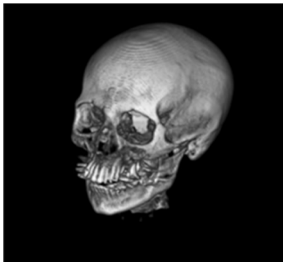


图 4 改进光线投射算法

表 1 文中算法与传统方法绘制时间比较 s

绘制算法	绘制时间	旋转时间
传统光线投射算法	5.23	5.06
基于包围盒算法	3.96	3.83
基于光线跳跃算法	3.04	2.94
改进的光线投射算法	1.89	1.83

根据表 1 的结果显示,相较于传统光线投射算法,基于包围盒的光线投射算法、基于光线跳跃的光线投射算法和文中改进的光线投射算法在绘制时间和旋转时间上分别有 1.32 倍、1.73 倍、2.76 倍的提高,可见文中改进的算法绘制速度最快。

4 结束语

文中提出了一种基于碰撞检测包围盒技术和光线跳跃的加速算法。首先利用包围盒技术隔离体数据周围无效的空体素,减少投射光线的数目;然后利用光线跳跃技术,在包围盒内快速进行光线的合成,跳过透明的空体素,减少大量体素的重采样。实验结果表明在图像质量不受影响的前提下,与传统光线投射算法、基于包围盒算法和基于光线跳跃算法相比,文中改进的算法在很大程度上减少了体素采样的计算,提高了绘制速度,同时又不降低图像质量。未来工作希望能改进包围盒技术和进行动态步长大小的设置,以适应不

同形状大小的体数据,进一步完善算法。

参考文献:

[1] 唐泽圣. 三维数据场可视化[M]. 北京:清华大学出版社, 1999.

[2] Lakare S, Kaufman A. Light weight space leaping using ray coherence[C]//Proc of visualization. [s. l.]; IEEE, 2004: 19-26.

[3] 樊 鹏,郭宝龙. 一种改进求交的自适应光线投射体绘制算法[J]. 计算机工程与应用, 2008, 44(4): 70-72.

[4] Tao Ling, Qian Zhiyu. An improved fast ray casting volume rendering algorithm of medical image[C]//Proc of 4th international conference on biomedical engineering and informatics. Shanghai; IEEE, 2011: 109-112.

[5] 何 晶,陈家新,黎 蔚. 基于 GPU 的实时光线投射算法[J]. 计算机工程与应用, 2008, 44(9): 40-42.

[6] 周志光,陶煜波,林 海. 一种有效显示隐藏特征的光线投射算法[J]. 计算机学报, 2011, 34(3): 517-525.

[7] 梁承志,高新波,邹 华,等. 空间跳跃加速的 GPU 光线投射算法[J]. 中国图象图形学报, 2009, 14(8): 1684-1688.

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

[9] 边园飞,杨 光,杨红雨. 改进的飞行模拟碰撞检测算法研究[J]. 计算机技术与发展, 2013, 23(1): 5-8.

[10] 赵奇峰,郭宝龙,张 磊. 基于 VTK 的医学图像可视化三维重建[J]. 计算机技术与发展, 2009, 19(6): 39-42.

[11] Hubbard P M. Approximating polyhedra with spheres for time-critical collision detection[J]. ACM Transactions on Graphics (TOG), 1996, 15(3): 179-210.

[12] 杨 晓,廉静静,张新宇. 基于 OSG 的虚拟场景中包围盒碰撞检测的研究[J]. 计算机技术与发展, 2011, 21(9): 32-34.

[13] Mascolo S, Casetti C, Gerla M, et al. TCP westwood: bandwidth estimation for enhanced transport over wireless links [C]//Proceedings of the 7th annual international conference on mobile computing and networking. [s. l.]; ACM, 2001: 287-297.

[14] Çelebi Ö C, Cevik U. Accelerating volume rendering by ray leaping with back steps[J]. Computer Methods and Programs in Biomedicine, 2010, 97(2): 99-113.

[12] 郭其标. 一种综合网络管理方案及其实现[J]. 科技信息, 2007(20): 360-361.

[13] 张德文,徐孟春,杜晓宁. XML 技术在分布式网络管理系统中的研究与应用[J]. 信息工程大学学报, 2007, 8(2): 231-234.

[14] Kim Song-Kyoo. Stochastic design of enhanced network management architecture and algorithmic implementations[J]. A-merican Journal of Operations Research, 2013, 3(1): 87-93.

(上接第 66 页)

术, 2008, 41(9): 98-100.

[10] Cavalas D, Greenwood D, Ghanbari M, et al. Advanced network monitoring applications based on mobile/intelligent agent technology[J]. Computer Communications, 2000, 23(8): 720-730.

[11] 刘 波,罗军舟,李 伟. 大规模网络管理中的任务分解与调度[J]. 通信学报, 2006, 27(3): 64-72.

一种高效的光线投射体绘制算法

作者：[方军](#)，[房晓阳](#)，[FANG Jun](#)，[FANG Xiao-yang](#)
作者单位：[南京理工大学 计算机科学与工程学院, 江苏 南京, 210094](#)
刊名：[计算机技术与发展](#)[ISTIC](#)
英文刊名：[Computer Technology and Development](#)
年，卷(期)：2014(8)

引用本文格式：[方军](#).[房晓阳](#).[FANG Jun](#).[FANG Xiao-yang](#) 一种高效的光线投射体绘制算法[期刊论文]-[计算机技术与发展](#) 2014(8)