

# 四种体绘制算法的分析与评价

尹学松<sup>1</sup> 张 谦<sup>1</sup> 吴国华<sup>1</sup> 潘志庚<sup>2</sup>

(杭州电子工业学院虚拟现实与多媒体实验室 杭州 310037)

(浙江大学 CAD&CG 国家重点实验室 杭州 310027)

E-mail: ynsyxs@hotmail.com

**摘 要** 体绘制技术是科学计算可视化的重要组成部分,具有较大的研究价值和广阔的应用前景。体绘制有两种方法即间接体绘制和直接体绘制,直接体绘制(简称体绘制)以其在体数据处理及特征信息表现方面的优势,已经得到了研究者越来越多的重视。文章对四种典型的体绘制算法进行了描述,并概述了它们的改进之处,同时对它们各自的性能进行了分析和评价。

**关键词** 体绘制 光线投射法 抛雪球法 错切-变形法 三维纹理映射法

文章编号 1002-8331-(2004)16-0097-04 文献标识码 A 中图分类号 TP301.6

## Review on Four Volume Rendering Algorithms

Yin Xuesong<sup>1</sup> Zhang Qian<sup>1</sup> Wu Guohua<sup>1</sup> Pan Zhigeng<sup>2</sup>

(Virtual Reality and Multimedia Laboratory, HZIEE, Hangzhou 310037)

(The State Key Laboratory of CAD & CG, ZJU, Hangzhou 310027)

**Abstract:** Volume rendering technique is one of important parts for ViSCC (Visualization in Scientific Computing). There are two classes: indirect volume rendering and direct volume rendering (simply called volume rendering). This paper discusses four widely-used volume rendering algorithms: raycasting, splatting, shear-warp and hardware-assisted 3D texture-mapping and gives a summarized description to improvements to each algorithm. In the end, it compares the pros and cons of these algorithms in order to promote further research of volume rendering algorithms.

**Keywords:** volume rendering, ray-casting, splatting, shear-warp, hardware-assisted 3D texture-mapping

### 1 引言

当今,科学计算可视化的研究与应用在发达国家可谓是如火如荼,一方面是因为它可以将大量抽象的数据或计算结果用图形或图象形象、直观地表示出来;另一方面是因为它的应用领域十分广泛,几乎涉及自然科学及工程技术的一切领域。作为科学计算可视化的一个重要组成部分——体绘制,自然得到了研究人员的高度重视。

体绘制研究始于 70 年代中期,它是伴随着断层投影(CT)、核磁共振(MRI)、超声波等医学成像技术的产生和发展而发展起来的。同时,它是以图象处理、计算机视觉和计算机图形学等学科为基础,借助于三维的基元(体素)将三维数据场的离散数据用二维的图象显示在屏幕上。它的优点是不仅能显示高质量的图象,更为重要的是能够比较清晰地显示出物体的内部结构,这是传统的计算机图形绘制方法所无法做到的,因此,体绘制研究深受人们的青睐。80 年代末,随着 Drebin 和 Levoy<sup>[1]</sup>等人提出了直接体视算法,有关体绘制研究日趋活跃。相应地,体绘制的应用范围从医学领域扩展到生物学、物理学、地理学、工业检测、气象学等领域。

在体绘制的发展过程中,计算机硬件性能始终是一个关键的制约因素,尽管现在计算机的计算速度、内存容量以及其它

图象加速设备的发展日新月异,但是对于庞大的体数据,体绘制速度始终难尽如人意。为了解决这种状况,研究人员们基于现有的硬件环境,根据各种应用的具体需求,在生成的图象质量和绘制速度之间寻求可能的最佳方案,使绘制速度尽量满足用户的要求,同时保证一定的图象质量,由此,形成了许多不同的体绘制算法。

### 2 四种典型的体绘制算法

体绘制按照处理对象的不同,可以分为对三维空间规则数据场的体绘制和对三维空间不规则数据场的体绘制。所谓不规则数据场指的是结构化数据场中的不规则数据和非结构化数据,象有限元分析及计算流体力学所生成的数据一般都属于这一类。在这类数据场中,体素的形状不同,大小不一,因而导致各种算法的效率降低。迄今为止,研究和开发三维不规则体数据的可视化算法仍然是一个有待进一步解决的问题。而对于规则数据场的体绘制研究趋于成熟,它有四种常用的算法:光线投射法(Ray Casting)、抛雪球法(Splatting)、错切-变形法(Shear-warp)和基于硬件的 3D 纹理映射方法(Hardware-assisted 3D texture-mapping)。下面就这四种算法及其优化方法做逐一介绍。

**作者简介:** 尹学松(1975-),杭州电子工业学院在读硕士研究生,研究方向是图象处理和体数据可视化。张谦(1972-),杭州电子工业学院在读硕士研究生,研究方向是图象处理和体数据可视化。吴国华(1970-),博士,副教授,研究方向是图象处理、体数据可视化和 PACS 体系结构。潘志庚(1965-),研究员,博士生导师,研究方向为虚拟现实、分布式图形、汉字信息处理及多媒体等。

## 2.1 光线投射法

光线投射法是一种以图象空间为序的体绘制方法,它从图象空间的每一像素出发,按视线方向发射一条射线,这条射线穿过三维数据场,沿着这条射线选择 $K$ 个等距的采样点,并由距离某一采样点最近的8个数据点的颜色值和不透明度值作三次线性插值(如图1),求出该采样点的不透明度值和颜色值。再将每条射线上各采样点的颜色值和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,从而可以在屏幕上得到最终的图象。

这种算法要遍历每个体素,而且,当观察方向发生变化时,数据场中的采样点之间的前后关系也必然变化,这样就要进行重新采样,计算量极为庞大,针对算法所存在的问题,人们提出了不少优化方法。

### (1) 光线提前终止

光线提前终止<sup>[2-4]</sup>是提高体绘制效率的一个非常有效的加速技术,它是按照从前到后的次序来跟踪从像素发出的每一条射线。当一条射线在穿过某个体素时,它的阻光度就进行累积,一旦阻光度达到了预先设定的门限值,就立刻停止射线的传播,这样,采样点后面的体素就被遮蔽了。该方法就是放弃被遮蔽的体素,保存采样得到的有用的体素来减少采样的区域,从而减少了计算量。

### (2) 空间数据结构

对于一个给定的可视化体数据集,一些体素簇能提供有用的图象数据,而另一些是无关的。空间数据结构的作用就是将相关一致的有用数据进行编码,从而有效地精选出不相关的体素。这些空间数据结构包括:金字塔结构、八叉树结构和 $k$ -d树等。空间数据结构<sup>[5]</sup>通过研究数据内部相关性来提高体绘制的效率,即允许从像素发出的射线跳过对绘制图象无关紧要的体素或者相对均匀的区域,从而达到减少计算量的目的。

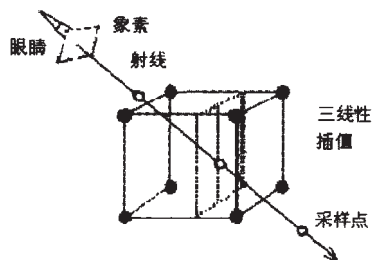


图1 光线投射法

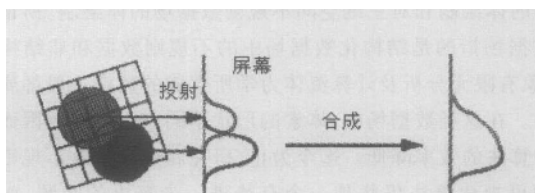


图2 抛雪球法

## 2.2 抛雪球法

与光线投射法不同,抛雪球算法<sup>[6]</sup>是反复对体素进行运算。它用一个称为足迹(Footprint)的函数计算每一体素投影的影响范围,用高斯函数定义强度分布(中心强度大,周边强度小),从而计算出其对图象的总体贡献,并加以合成,形成最后的图象。由于这个方法模仿了雪球被抛到墙壁上所留下的一个扩散状痕迹的现象,因而得名“抛雪球法”(见图2)。

因为抛雪球算法是“以物体空间为序”的体绘制算法,所以它的优点就是能按照体数据存储顺序来存取对象,同时只有与图象相关的体素才被投射和显示,这样可以大大减少体数据的存取数量<sup>[7]</sup>。而且算法适合并行操作。

但对于向前投射算法,象平面上的投影面是随着视点改变而随意缩放和旋转的,因而精确计算对周围像素影响的范围和对每一像素所影响的大小是十分费时的。在透视投影中,这种变换更是按每一体素而不同。同时重构函数的选择也要十分仔细,要避免由于选择不当而产生的相邻体素投影到象平面上而产生的间隙或过分重叠。正因为这个原因,设计一个既能保证图象质量又是高效的重构函数往往比较困难。

从理论上说,使用同样的重构函数权值,抛雪球算法能够生成和光线投射算法相同质量的图象。但在实际应用中,由于其权值计算比较困难,所以经常使用近似算法,因而图象质量会有所下降。

另一种抛雪球算法运用了2D图象弯曲技术<sup>[6]</sup>。其主要思想是:将体数据分割成许多二维数据层,应用二维弯曲技术将每一层变换到象空间中,并将处理后的数据层合成为最终的图象。运用二维滤波器使算法的复杂度大大下降,但它的投影面计算和再取样权值的计算仍然和视角有关,所以不能在保证图象质量的同时用软件快速实现。这种方法会出现发光面振荡的现象。K Mueller和R Crawfis<sup>[8]</sup>提出了基于图象序列的抛雪球方法,以消除这种现象。

抛雪球法存在一个和光线提前中止相似的优化方法:投射提前中止。其过程是:将每一相关体素变换投影到象平面,同时将其索引映射到阻塞表中(该表中保存象平面上每点的索引值及可见性标识等),当检测该体素所对应的表项为可见,即完成所有操作,当其为不可见则中止该体素的绘制。它与光线提前中止不同的是:后者在光线中止之后,将忽略后序的体素。而投射提前中止对每一个体素的投影是必不可少的,只是省略了对不可见体素“足迹”函数的计算及光栅化。J.Orchard,T.Möller<sup>[9]</sup>提出了一种利用3D相邻数据结构的加速绘制方法。

## 2.3 错切-变形法

平行投影的错切-变形算法最早是由Cameron和Undrill<sup>[10]</sup>提出的,其基本思想包括3个步骤(见图3):

- (1) 将体数据变换到错切后的物体空间,并对每一层数据重采样;
- (2) 按从前至后的顺序将体数据投射到二维中间图象平面;
- (3) 通过变形变换,将中间图象投射到象空间产生最终的图象。

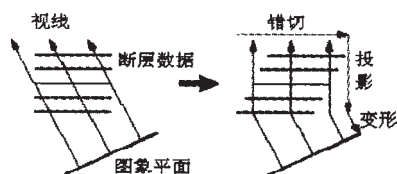


图3 平行透视错切-变形算法

Lacroute和Levoy<sup>[11]</sup>对算法进行了优化。由于遍历中间图象的扫描线平行于遍历体素的扫描线,所以可以应用基于扫描线的数据相关特性,通过预处理,对体数据按定义的不透明度进行编码,将数据分为透明和不透明两种,利用连续区域编码技

术,将扫描线分割为一段段透明和不透明相间的区域,在绘制过程中,可以跳过透明区域,而只处理不透明的体素。同时对中间图象定义一种数据结构保存每一不透明象素按扫描线方向和它最相近的透明象素之间的偏移量。对中间图象扫描时掠过与其相对应的透明体数据区域和自身不透明的象素区域,从而减少计算量,提高绘制速度。

由于算法中的体数据层缩放比例一致,可以选一个特定的比例因子,使每一体数据扫描线存在着从体素到象素的一一对应的关系,利用这个关系,在处理不透明象素时,可以采用双线性插值和聚类卷积方法。在处理光照模型时,Lacroute和Levoy<sup>[11]</sup>采用了基于查找表的体系结构。对于变形变换,由于它是对二维图象进行处理,相对于体数据运算,其计算量较小,可采用双线性过滤器的仿射变换。

在对平行透视算法优化的同时,Lacroute和Levoy<sup>[11]</sup>还提出了透视投影的错切-变形算法和利用最大-最小八叉树快速分类方法的错切-变形算法,它们被认为是现在最快的软件体绘制方法。

这类算法属于采用多遍重采样的体绘制技术的方法,它与同种算法不同的是,这种方法只要求两次重采样过程(其他算法要求三次或更多),并且第二次是二维的弯曲变换,对体素的遍历只有一次,计算量大大减少。但该算法也存在着一些不足,如视角为45°时将出现阶梯状走样,图象放大后变模糊等,J.Sweeney和K.Mueller<sup>[12]</sup>提出了改进的方法。

2.4 基于硬件的三维纹理映射

绘制的速度一直以来都是体绘制的主要问题,因为对生成的图象来说,物体的每个区域的处理都会影响到它的质量,因此需要计算每个体素,而且体素化以后,重新定位体素又需要相当大的计算量。为了解决体绘制速度的问题,人们提出了许多改进的算法,这些算法的共同点就是在软件的基础上来提高绘制的效率,如上面介绍的算法。而三维纹理映射方法是基于硬件来提高体绘制的速度。

所谓基于硬件的三维纹理映射指的是,在纹理空间中实现重采样的插值运算及具有不透明度值的图象合成等操作均由硬件完成,从而大大提高了运算速度。这种方法首先将体数据作为三维纹理图,装入纹理内存,然后在体数据内部定义一系列采样多边形去采样物体的纹理,再通过查找表将采样得到的数据转换为相应的颜色值及不透明度值,这样就可以按照从后向前的顺序进行图象合成,投影于视平面而形成最后的图象。

为了使生成的图象具有明暗效果,提高图象的质量, Frank Dachille<sup>[13]</sup>等人提出了一种基于三维纹理映射硬件支持的具有明暗效果的体绘制方法。他们的方法是先将密度数据和梯度数据装入纹理内存中,然后用纹理硬件沿着光线投射的方向对它们进行重采样,再把重采样数据变换到内存中的缓存器里保存,最后,通过CPU对这些数据进行明暗处理与合成,形成每个象素的RGB值,存储到缓存器里,利用硬件就可以显示出图

象。他们的方法是用随视线变化的光线来对每个可见的采样点进行明暗处理,所以能绘制出高质量的图象。M.Meißner<sup>[14]</sup>等人也提出了一种具有明暗效果的算法,该算法对插值的数据进行了分类,并使用OpenGL进行绘制,因此提高了绘制速度。

在上述基于硬件的三维纹理映射算法的实现中,出现了一个问题,即计算机的纹理内存是有限的,因而使得大规模体数据在绘制时,必须被分成若干子数据块,依次装入纹理内存中进行绘制,由此引发的频繁的输入/输出操作及重采样运算使大规模体数据的体绘制性能急剧下降。针对这一问题,童欣<sup>[15]</sup>等人提出的方法是利用空间跳跃技术,来有效地降低纹理内存的数据交换量,进一步加速了体数据的绘制。他们又提出了一个有效的基于纹理硬件加速的大规模体数据的快速绘制方法<sup>[16]</sup>。这种方法与原始的基于纹理硬件的体绘制方法不同,采用了4级体数据装入流水线,从而使体数据在绘制前得到有效的预处理。同时,通过使用索引结构加速体数据分类,这样将最终结果图象有贡献的体元装入纹理内存并用于绘制,从而有效地降低了系统负载。

3 体绘制算法的比较

光线投射法最主要的问题就是不能按照物理存储的顺序来存取体数据,因为视线方向是任意的,导致视线与体素可以以任意方向相交。任何与视点相关参数的改变,都会引起光线投射算法的完全重绘制。这样就不得不花大量的时间去计算采样点的位置,及其属性值。光线投射法存在的另一个问题是对内存的需求量大。由于算法对数据存取的任意性,使具有空间数据结构的体数据阵列无用武之地。改进的光线投射法如光线提前终止和空间数据结构等算法很好地减少了计算量,提高了绘制速度。

抛雪球算法能按照体数据存储顺序来存取对象,同时只有与图象相关的体素才被投射和显示,可以大大减少体数据的存取数量。算法也适合并行操作。但当观察方向发生变化时,要重新计算重构函数空间卷积域在平面上的投影区域,并且要对投影区域中相对应的每个体素重新进行旋转和比例变换,这样计算量也相当大。

剪切-变形法是将三维离散数据场的投影变换分解为三维数据场剪切变换和二维图象的变形两步来实现,从而将三维空间的重采样过程转换成二维平面的重采样过程,大大减少了计算量。使得三维数据场的体绘制可以在图形工作站上以接近实时的速度实现,而不显著降低结果图象的质量。但该方法有个局限就是三维数据场的数据在向中间坐标系(错切物体空间)变换时,观察方向必须与三维坐标系中的某一轴重合,否则它的优点不复存在。

基于硬件的三维纹理映射是在纹理空间中实现重采样的插值运算及具有不透明度值的图象合成等操作均由硬件完成,从而大大提高了运算速度。而且当形体表面的细节十分复杂而

表1 典型算法比较<sup>[17]</sup>

算法	速度	特点
光线投射法	体素寻址困难,速度慢	内存开销大
抛雪球法	体素寻址快捷,加速技术多样,速度较快	可实现渐进显示,内存开销小
错切变形法	继承抛雪球法寻址加速的优势,且速度更快	通过二维形变简化体数据旋转变换,内存开销小
维纹理映射	利用硬件处理数据,绘制速度较快	算法中的重采样以及颜色和不透明度的值计算由硬件完成,因此内存开销大



精细时,对其进行体绘制,这种方法显得尤为重要了。当然硬件环境的性能是这种方法的关键。

#### 4 相关工作的结果

该文利用错切-变形算法对医学 CT 图像(膝部骨骼)进行了三维重建,结果如图 4,从结果可以看出利用体绘制方法重建出来的图象比较完好地保留了物体的真实细节,同时绘制速度也相对比较快。

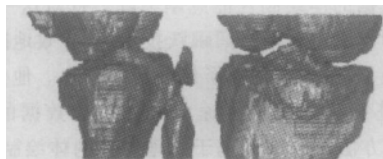


图 4 膝部骨骼三维重建图

#### 5 结束语

文章对体绘制的概念作了简要的介绍,并对四种典型而又常用的算法及其它们改进的算法进行了分析和比较,由此,可以看出尽管有多种不同的体绘制算法,但其实质均为重新采样与图象合成。同时,也不难发现重新采样是体绘制算法中最费时的部分,也是计算量最大的部分。如果要提高体绘制的效率,减少重新采样的计算量是行之有效的好方法。

近年来,在体绘制领域出现了一些引人瞩目的新研究方向,它们包括向量体数据和高维体数据可视化问题,以及多种体数据的融合等。同时,关于几何表示和体素表示混合体视的研究逐渐引起了人们的兴趣,使传统计算机图形学和体绘制的结合更为紧密。可以相信,体绘制作为一种技术将会得到飞速发展,并且在各个领域发挥重要的作用。(收稿日期:2003 年 9 月)

#### 参考文献

1. M Levoy. Display of Surfaces from Volume Data[J]. IEEE Computer Graphics and Applications, 1988, 8(3): 29~37
2. D Bartz, M Meißner. Voxels versus polygons: a comparative approach for volume graphics[C]. In: Proc Volume Graphics'99, Swansea, 1999: 33~48
3. K Kwansik, C Wittenbrink, A Pang. Maximal-abstract differences for comparing direct volume rendering algorithms[R]. Hewlett-Packard

Research Lab Technical Report 2000-40

4. H Pfister et al. The VolumePro real-time ray-casting system[C]. In: Proceedings of SIGGRAPH '99, 1999: 251~260
5. M Meißner et al. A practical evaluation of four popular volume rendering algorithms[C]. In: ACM Symposium on Volume Visualization, 2000
6. L Westover. Footprint evaluation for volume rendering, In Computer Graphics Proceedings[C]. In: Annual Conference Series ACM SIGGRAPH, 1990: 367~376
7. K Mueller et al. High-quality splatting on rectilinear grids with efficient culling of occluded voxels[J]. IEEE Transactions on Visualization and Computer Graphics, 1999, 5(2): 116~134
8. K Mueller, R Crawfis. Eliminating popping artifacts in sheet buffer-based splatting[C]. In: Proc Visualization'98, 1998: 239~245
9. J Orchard, T Möller. Accelerated Splatting Using a 3D Adjacency Data Structure[C]. In: Graphics Interface 2001, 2001: 191~200
10. G Cameron, P E Undrill. Rendering volumetric medical image data on a SIMD-architecture computer[C]. In: Proceedings of the Third Eurographics Workshop on Rendering, Bristol, UK, 1992: 135~145
11. P Lacroute, M Levoy. Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transform[C]. In: Computer Graphics, Annual Conference Series, 1994: 451~458
12. J Sweeney, K Mueller. Shear-Warp Deluxe: The shear-warp algorithm revisited[J]. Joint Eurographics-IEEE TCVG Symposium on Visualization 2002, Barcelona, Spain, 2002: 95~104
13. F Dachille, K Kreeger, B Chen. High-Quality Volume Rendering Using Texture Mapping Hardware. 1998 SIGGRAPH/Eurographics Workshop on Graphics Hardware, 1998: 69~76
14. M Meißner, U Hoffman, W Straßer. Enabling classification and shading for 3D texture mapping based volume rendering[C]. In: Proc Visualization'99, 1999: 207~214
15. TONG Xin, TANG Zesheng. Hardware-assisted 3d texture mapping volume rendering algorithms using spatial leaping[J]. Chinese Journal of Computer, 1998, 21(9): 807~812
16. TONG Xin, TANG Zesheng, XU Zhongxin. Efficiently rendering large volume data using texture mapping hardware[J]. Journal of TSINGHUA UNIVERSITY, 2000, 40(1): 72~75
17. SHEN Haige, KE You-an. Classification and review on 3-D visualization algorithms to medical volume data[J]. Journal of Image and Graphics, 2000, 5(8): 545~550

(上接 86 页)

方法。不仅体现在时间和空间上,而且体现在处理结果上。

利用多项式拟合算法时,首先用来作为拟合根据的点数不能太多。因为拟合函数是利用最小二乘法得到的,它要求拟合函数与用来作为拟合根据的各点差值的平方和达到最小。如果作为拟合根据的点数太多,拟合函数反映的是众多数据点,而不是针对端点处的变化趋势。在文中采用三个点来确定拟合函数,较好地反映了原数据序列在端点处的变化规律,有效地减小了误差。

其次多项式的次数也是值得商榷的。因为只利用了三个极值点作为拟合的根据,所以多项式的次数一次或两次为佳,次数太高就会导致失真,拟合函数不能反映出原数据序列在端点

处的变化趋势。(收稿日期:2003 年 8 月)

#### 参考文献

1. Huang N E et al. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis[J]. Proc R Soc Lond A, 1998: 454: 903~995
2. 邓拥军等. EMD 方法及 Hilbert 变换中边界问题的处理[J]. 科学通报, 2001, 46(3): 257~263
3. ZHAO Jin-Ping et al. MIRROR EXTENDING AND CIRCULAR SPLINE FUNCTION FOR EMPIRICAL MODE DECOMPOSITION METHOD[J]. Journal of Zhejiang University(SCIENCE), 2001, 2(3): 247~252
4. 杨大地等. 数值分析[M]. 重庆大学出版社