

# 基于预计算切片序列的动态体绘制技术

宫中樑, 唐 杰, 武港山

(南京大学 计算机科学与技术系, 江苏 南京 210093)

**摘要:**为进一步提高基于硬件加速的体绘制技术的渲染性能,解决海量体绘制应用中体数据可视化环节的性能瓶颈,文中提出了一种基于硬件加速的实时切片体绘制技术的优化模型。相对于完全基于实时切片的体绘制系统,应用该技术在保证最终图像渲染质量的同时可节省动态判断切片方程的计算开销并大幅降低运行时的数据带宽需求。基于预计算的动态体绘制在切片方向选择空间中选取一组彼此正交的切片方案,并在预处理阶段生成切片几何数据,运行时根据视点位置通过特定判断算法进行方案选择。对基于分块的海量数据体绘制系统,该技术可有效降低海量体绘制的总体带宽消耗,从而进一步拓展系统的渲染规模。

**关键词:**体绘制;切片;海量数据;图形处理单元

**中图分类号:**TP391.41

**文献标识码:**A

**文章编号:**1673-629X(2010)08-0005-04

## A Precomputed Slicing Based Volume Rendering Technique

GONG Zhong-liang, TANG Jie, WU Gang-shan

(Department of Computer Science and Technology, Nanjing University, Nanjing 210093, China)

**Abstract:** To improve the rendering performance and solve the major bottleneck of the mass-data volume rendering system a new slice based hardware accelerated volume rendering technique is given in this paper. Comparing with the common real time slicing technique, the new version can decrease the CPU cost as well as the runtime data bandwidth which is critical to the volume rendering system. The new technique generates a set of slicing data on the initialization stage and when the volume needs to be rendered the system just chooses an appropriate one from the precomputed slicing data. For mass-data volume rendering system, the technique can dramatically decrease the bandwidth requirement and so that larger dataset can be rendered.

**Key words:** volume rendering; slicing; mass data; GPU

## 0 引言

基于硬件加速的体数据可视化技术<sup>[1]</sup>主要分为两类:一为基于图像序列的切片绘制技术<sup>[2]</sup>,二为基于屏幕空间的光线投射技术<sup>[3]</sup>。光线投射方式可获得较好的图像质量,且在多实体绘制时有更好的可控性<sup>[4]</sup>,但因其基于图像空间的射线探测,因此绘制的基准开销很大且对图形硬件的规格有较高要求,尤其对于基于分块的海量体绘制领域,光线投射方式仍然缺乏灵活性;基于图像序列的切片体绘制方式是利用一系列2D图像的渲染产生体的效果,切片体绘制对硬件要求低且可完全利用硬件光栅化能力,因此长期作为基于硬

件加速的实时体绘制领域的首选方案。此外,切片技术不只用于对数据体的可视化,在数据集的采集与重构过程中也常用到相关切片技术<sup>[5]</sup>。

基于图像序列的动态切片算法在每次重绘时,都需要对待绘制数据体重新切片,并将生成的切片几何数据传给图形硬件进行绘制。这种动态切片方式可保证生成的切片序列在当前视点位置下能够产生最好的图像质量,但其需在运行时对数据体的包围盒进行实时切片计算,且每次重绘都需向GPU传输切片序列的实际几何数据,这在实时绘制应用中对CPU和数据总线都会造成很大的负担<sup>[6]</sup>。

学术领域一直在探究利用图形硬件增强体绘制效率并利用有限资源实现大规模渲染<sup>[7]</sup>的方式。1999年Lamar首次提出利用硬件加速结合八叉树实现体绘制技术<sup>[2]</sup>,打破了有限资源下渲染大规模体数据的限制。因受制于当时的图形硬件规格,其基于二维纹理的实现方式并不高效,后续有许多改进工作来适应图形硬件的发展和进一步扩展数据渲染规模<sup>[8]</sup>,但其本

收稿日期:2009-12-19;修回日期:2010-02-01

基金项目:江苏省自然科学基金(BK2008262);江苏省高新技术研究计划(工业)(BG2007037)

作者简介:宫中樑(1984-),男,山东青岛人,硕士研究生,研究方向为海量地质数据可视化;唐 杰,副教授,博士,研究方向为计算机图形学;武港山,教授,博士,研究方向为多媒体信息检索。

质依然是依靠高级的场景管理进行更加高效的数据调度或快速的层次化裁剪。国内也有部分体绘制优化工作在近年被提出,如薛健等人基于海量医学数据处理框架提出的体绘制算法<sup>[9]</sup>,肖永飞等人提出的基于硬件加速的大数据量自适应体绘制<sup>[10]</sup>,宋涛等人提出的快速体绘制<sup>[11]</sup>以及马晓晨等人的多分辨海量体绘制<sup>[12]</sup>等。

文中着眼于基于硬件加速的实时体绘制机制本身,针对基于图像序列的动态切片算法的特点,提出一种新的基于预计算切片的动态判断体绘制技术。基于预计算的动态判断体绘制无需任何运行时切片计算或几何数据传输,允许不同数据分块间共享预计算结果,数据存取代价低,在不增加过大存储负担的情况可大幅提升绘制系统渲染性能。

## 1 运行时切片的必要性研究

基于图像序列的体绘制技术仅是一种近似模拟体数据场的手段,其在整个体数据场上并非均匀采样,当切片方向与视线近似垂直时,因切片间隔与面片内光栅化密度不平均会产生严重的视觉走样。因此传统动态切片体绘制算法需要在场景更新阶段对数据体进行重新切片,以保证最终图像不会出现因采样密度不均导致的明显视觉走样。通常,当视线方向与切片方向平行时视觉效果最佳,当视线方向与切片方向垂直时则最差。

文中基于预计算的动态体绘制技术的提出基于如下观察事实:

(1) 视线与切片方向的角度对于最终图像质量的影响并非线性关系。

(2) 只当视线与切片方向近似垂直时,才会出现视觉走样。

(3) 每一种切片方向都存在一个相当广的可视角范围,最终图像质量对于视线在此范围内的变化并不敏感。

设视线单位向量为  $e$ ,切片方向单位向量为  $s$ ,这里假设  $e$  与  $s$  的夹角为非锐角,若夹角为锐角则反转  $s$ 。这样,对于切片导致视觉走样的问题可以描述为关于  $e$  与  $s$  间夹角变化的问题。 $e$  与  $s$  间夹角对于视觉走样的上限阈值,这里无法给出确定值,且此上限往往与执行光栅化的具体硬件相关。但可基于实验给出一个不会出现视觉走样的下限,设此下限为  $\epsilon$ ,对于  $\epsilon$ ,根据其含义可推得只要  $e, s$  满足如下条件则不会出现视觉走样:

$$\text{dot}(e, s) = \cos(e \cdot s) < \cos(\epsilon) \quad (1)$$

不妨设规则数据体的外包围框为一立方体,切片

计算可理解为将此立方体沿某一方向(即向量  $s$  所代表的方向)进行切片,沿不同的方向可能产生不同的多边形切片序列。按照前述理论,若能在整个切片方向空间中预选取一组切片方案,且保证在任何时刻,这组切片方案中至少有一种方案是符合公式(1)的,则在任何时刻这组预选取的切片方案中至少有一种方案是可用的。因此,基于预计算的切片方案存在理论上的可行性。

## 2 基于预计算切片的动态体绘制技术

### 2.1 对规则数据体的预切片计算

文中的预计算方案选取三维空间中三个标准坐标轴向量作为切片方向,鉴于对每个坐标轴,都有正负两种方向,所以共有六种切片方案。

对于这种切片抽样方式,记  $m$  为最差情况下向量  $e$  与  $s$  的夹角,则  $m$  满足式(2):

$$m = 180 - a \cos(\text{dot}(<1, 0, 0>, \frac{<1, 1, 1>}{\|<1, 1, 1>\|})) \quad (2)$$

$$m \approx 125$$

只要证明  $m > \epsilon$ ,则这种切片方案就不会遇到明显视觉走样。但  $\epsilon$  是一个与实际环境问题相关的参数值,因此无法给出严格证明。在文中工作遇到的所有实验环境中,  $\epsilon$  都远远小于  $m$ ,因此有理由相信,这种标准轴向的抽样方案具备可用性和普遍适用性。

需要注意的是,切片方向的采样方案与其处于何种意义的空间无关,根据  $m$  的定义,任取向量  $e$ ,都存在一种切片方向为  $s$  的方案,满足公式(3):

$$\cos(e \cdot s) \leq \cos(m) < \cos(\epsilon) \quad (3)$$

因此,不妨设切片方向的采样方案处于数据体的局部空间中,且立方数据体的各边分别平行于三个坐标轴。这一方面保证了切片产生的所有几何面片都为标准长方形,同时简化了切片的计算,实际切片算法只需按照固定步长延每个轴向采样即可。上述切片方案降低了计算开销,同时节省了将非标准多边形面片转化为标准四边形或三角形的时间。

一个完整的基于标准轴向切片算法伪代码如下:

```
init slice_data[6]; init slice_step;
for(float i = .0f; i < cube_len; ++slice_step){
    quad_x.p0 = <i, 0.0, 0.0>;
    quad_x.p1 = <i, 1.0, 1.0>;
    quad_x.n = <1.0, 0.0, 0.0>;
    quad_ix.p0 = <cube_len - i, 0.0, 0.0>;
    quad_ix.p1 = <cube_len - i, 1.0, 1.0>;
    quad_ix.n = <-1.0, 0.0, 0.0>;
```

```

add quad_x to slice_data[0];
add quad_ix to slice_data[1];
perform slicing for y & z coordinate just like x;
}

```

## 2.2 动态判断体绘制算法

根据上节所述,可以在预处理阶段生成一组固定的切片方案预存于图形硬件的显存中,当数据体需要重绘时,只需根据当前的视点与数据体的位置关系从预处理好的切片数据中选择一种不会出现走样的方案即可。基于预计算切片的动态体绘制运行流程如图1所示。

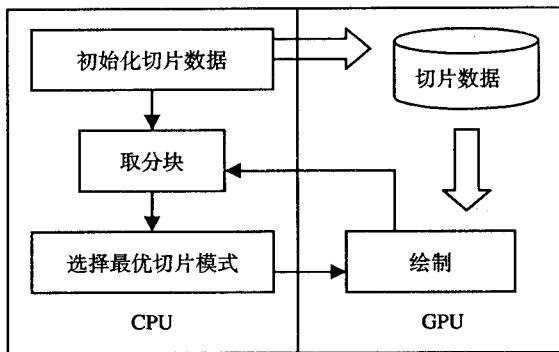


图1 基于预计算切片的动态体绘制流程图

由图1可知,切片生成的几何数据在初始化阶段被一次读入显示设备存储器中,重绘时,CPU仅对当前各数据分块做判断选择,并将选择结果由总线传输给GPU进行绘制,这样的运作模式使得切片体绘制的运行时数据带宽需求大幅降低,且动态裁剪的步骤也被极大地简化。对于每个数据体,这里选取其中心点 $o$ 作为参考点,则视线方向向量 $e$ 的计算方式如公式(4)所示:

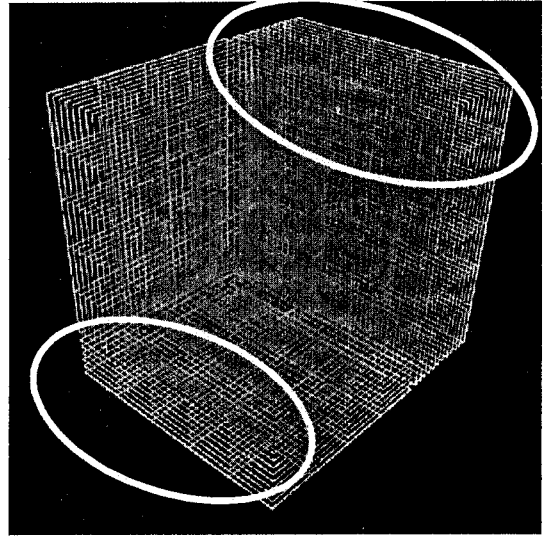
$$e = \frac{o - \text{eye\_pos}}{\|o - \text{eye\_pos}\|} \quad (4)$$

取数据中心点作为参考点存在一个弊端,当相机处于数据体内部时,有可能获得错误的视线方向,这种情况需要特别处理,或修改参考点为数据体的最远端。

动态判断算法并不引入对 $\epsilon$ 的直接判断,因为正确性问题在方案选择时已给予保证,判断算法只需遍历所有备选方向,找出最接近动态切片方案的预生成方案,设 $\text{ret\_s}$ 为最终中选方案的切片方向, $M$ 为世界空间到数据体局部空间的正交变换矩阵,则 $\text{ret\_s}$ 满足公式(5):

$$\begin{aligned} \text{ret\_s} &= \arg \min_s \text{dot}(s, M \times e) \\ s &\in \text{PrecomputedDets} \end{aligned} \quad (5)$$

图2为一个基于分块的动态判断实例,图中每个分块的切片方向都根据其自身与视点的位置关系单独判断。



(注意图中圈中的区域与中央区域的切片方向并不相同)

图2 采用动态切片判断的实例

## 3 实验结果与分析

基于文中的工作,实现了一套基于八叉树的海量数据多分辨率绘制引擎<sup>[5,11]</sup>,系统基于多分辨率的分块控制并采用了基于预计算切片序列的动态体绘制技术。为达到实验对比目的,系统同时搭载了传统动态切片体绘制技术。

实验平台为 Intel Xeon 2.8GHz x 2, 1G 内存, Geforce9800GTX 512M 显卡, Windows XP, 开发平台采用 VS2005, 图形开发接口使用 OpenGL。

表1列出了体绘制系统处理不同体积数据时的实际性能,绘制系统采用限定最高帧率为30FPS的策略以避免过度绘制抢占计算和带宽资源。为模拟实际情况,每隔一定时间会向绘制系统发送更新或重绘命令使系统保持繁忙状态。

表1 交互请求频繁情况下体绘制系统渲染性能列表

数据尺寸	体数据大小	平均帧率	传统切片模式帧率
352x352x256	216M	30.2	26.3
600x600x6000	10.8G	24.4	16.1
2000x2000x6000	100G	9.2	4.7

由表1可以看到,对于三种体积的数据集,采用文中算法相较传统切片模式性能皆有明显提升。基于预计算的体绘制系统可在交互操作频繁的情况下流畅渲染10G级数据集,在100G数据集上也可保持10帧左右的平均帧率。根据追踪帧率变化的曲线显示,在数据调度频繁的情况下,体绘制系统仍可保持稳定的性能,如图3所示。

实验结果表明,基于预计算切片序列的动态体绘制技术在保证最终渲染质量的同时,得益于其低计算

开销与低带宽需求的特点,使绘制系统的渲染规模得以进一步扩展,并大大缓解了体绘制阶段的性能瓶颈。

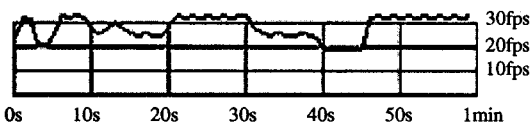


图 3 600x600x6000 数据集的 1 分钟帧率跟踪曲线

#### 4 结束语

提出了一种基于预计算切片序列的动态判断体绘制技术,该技术从切片方向空间中预选取一组切片方案作为备选方案,绘制阶段根据数据体与视点的位置关系从备选方案中选取最适合的切片方案进行绘制。该技术在运行时传输方案索引号而非实际几何数据,从而达到降低带宽需求的目的,同时,该技术还拥有运行时计算开销低,切片结果规则且易于处理等优点。通过理论推导结合实验反馈验证了该算法的普遍适用性。同时,实验结果证明了该算法可以达到提高体绘制性能、拓展体绘制规模的目的。

目前该技术也存在尚待改进之处。首先,该算法对非分块的数据集体绘制效果尚不理想,且不支持视点处于数据体内部的情形;另外,因几何数据基于预计算,步长固定,所以一定程度上减少了多分辨率控制的灵活性。

#### 参考文献:

- [1] Akeley K. Reality Engine Graphics[J]. Computer Graphics, 1993, 27: 109-116.
- [2] Lamar E, Hamann B, Joy K I. Multiresolution techniques for interactive hardware texture-based volume visualization

[C]//Proceedings of IEEE Visualization. Los Angeles, California:[s. n.],1999:355-361.

- [3] Gobbetti E, Marton F, Antonio J. A single pass GPU ray casting framework for interactive out-of-core rendering of massive volumetric datasets[J]. Vis Comput, 2008, 24(7): 797-806.
- [4] Lefebvre S, Dachsbacher C. Tiled trees[C]// Proceedings of SIGGRAPH. San Diego, California:[s. n.],2007: 25-31.
- [5] 马仁安,张二华,杨静宇. 不规则地质体的分割与体绘制方法研究[J]. 计算机研究与发展,2005,42(5):883-887.
- [6] 曹轶,莫则尧,王弘堃. 协同分布式图形硬件的混合并行体绘制[J]. 中国图像图形学报,2008,13(7):1379-1384.
- [7] Gobbetti E, Marton F. Far voxels: a multiresolution framework for interactive rendering of huge complex 3d models on commodity graphics platforms[C]// Proceedings of SIGGRAPH. Los Angeles, California:[s. n.],2005:878-885.
- [8] Crail C, Fabrice N, Sylvain L. GigaVoxels: ray-guided streaming for efficient and detailed voxel rendering[C]//Proceedings of the symposium on Interactive 3D graphics and games. Boston, Massachusetts:[s. n.],2009:15-22.
- [9] 薛健,田捷,戴亚康. 海量医学数据处理框架及快速提绘制算法[J]. Journal of Software,2008,19(12):3237-3248.
- [10] 肖永飞,付宜利,王国树. 硬件加速的大数据量自适应体绘制[J]. 计算机辅助设计与图形学学报,2009,21(5):612-616.
- [11] 宋涛,欧宗瑛,王瑜. 八叉树编码体数据的快速体绘制算法[J]. 计算机辅助设计与图形学学报,2005,17(9): 1990-1996.
- [12] 马晓晨,孔小利,陈建军. 一种大规模三维地震数据体绘制的 LOD 技术[J]. 大庆石油学院学报,2008,32(4):23-26.

(上接第 4 页)

terms[C]//In: Getoor L, Senator T E, Domingos P, Faloutsos C, eds. Proc. of the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Washington: ACM, 2003:286-295.

- [24] Huan J, Wang W, Prins J. Efficient mining of frequent subgraphs in the presence of isomorphism[C]// In: Kumar V, Tsumoto S, Zhong N, Yu P S, Wu X, eds. Proc. of the 2003 IEEE Int'l Conf. on Data Mining. Melbourne: IEEE Computer Society, 2002:549-552.
- [25] Wernicke S. Efficient detection of network motifs[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics,2006,3(4):347-359.
- [26] Kashtan N, Itzkovitz S, Milo R, et al. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs[J]. Bioinformatics, 2004, 20(11): 1746-1758.
- [27] Huan J, Wang W, Prins J, et al. Spin: mining maximal frequent subgraphs from graph databases[C]//Proceedings of the tenth ACM SIGKDD International Conference on Knowledge discovery and data mining. [s. l.]:[s. n.], 2004:581-586.
- [28] Baskerville K, Paczusi M. Subgraph Ensemble and Motif Discovery Using a New Heuristic for Graph Isomorphism[EB/OL]. 2006-06-19. arXiv.org, q-bio/0606023.
- [29] Ciriello G, Guerra C. A Review on Models and Algorithms for Motif Discovery in Protein-Protein Interaction Network[J]. Briefings in Functional Genomics and Proteomics,2008,7(2): 147-156.
- [30] Kuramochi M, Karypis G. Finding frequent patterns in a large sparse graph[C]// In: SIAM International Conference on Data Mining (SDM'04) 2004. SIAM, Lake Buena Vista, Florida, USA:[s. n.],2004:243-271.