

文章编号:1001-9081(2007)05-1129-03

基于标准 PC 机的大数据实时体绘制算法研究

谢凯¹, 杨杰², 朱跃敏³

- (1. 湖南大学 计算机与通信学院, 湖南 长沙 410082;
2. 上海交通大学 图像处理与模式识别研究所, 上海 200240;
3. 法国国家科学研究中心, 法国 里昂 69621)
(xie_kai2001@sina.com)

摘要:为了解决在标准 PC 机上对大数据进行实时体绘制的问题,提出了一种基于图形处理器的大数据高质量体绘制算法。该算法采用三维纹理映射作为核心的绘制算法,结合可见性测试、遮挡测试和模板测试来加快绘制速度。实验结果表明,对虚拟人体数据,可以在不损失图像质量的前提下,以可交互的速度进行绘制。

关键词:体绘制;三维纹理映射;图形处理器

中图分类号: TP391 **文献标识码:**A

Real-time volume rendering of large datasets on standard PC hardware

XIE Kai¹, YANG Jie², ZHU Yue-min³

- (1. School of Computer and Communication, Hunan University, Changsha Hunan 410082, China;
2. Institute of Image Processing & Pattern Recognition, Shanghai Jiaotong University, Shanghai 200240, China;
3. the French National Center for Scientific Research, Lyons France)

Abstract: A high quality volume rendering technique was presented to render a large volume dataset at interactive rates on standard PC hardware. We employed 3D texture mapping as a core rendering engine and took advantage of combinations of HW-supported visibility tests such as occlusion queries, stencil tests and programmable shaders to accelerate the whole rendering process. We have implemented the volume rendering algorithm for large volume data, and achieved real-time performance without any loss of image quality.

Key words: volume rendering; 3D-texture map; graphics processing units

0 引言

近年来,对大体数据进行绘制的需求在逐步增加,体数据获取设备产生了庞大的数据,如:一个 CT 扫描仪可以产生 1 500 多张切片的二维图像,分辨率为 512×512 。显卡技术的迅速发展使得直接体绘制达到可以交互的速度,一种典型的基于显卡的体绘制方法是利用显卡支持的三维纹理映射进行绘制。但是,显卡的有限显存不能对大的体数据进行直接体绘制。在文献中,为了解决大数据体绘制的问题,研究者通过压缩原始数据,然后在显存中解压,这种方法会损失部分图像质量^[1,2]。并且,这种方法在医学成像和科学可视化中是有问题的。如:在医学成像中,对病人的重要区域的显示由于数据的压缩而遗漏。这样就会误导医生产生不正确的诊断结果。而且,医生想在任何时候以高分辨率(如 512×512 窗口)显示体绘制的结果,这样,结果图像中的质量损失就变得非常明显,会导致对病人的错误诊断^[3]。

为了解决在标准 PC 机上对大数据进行实时体绘制的问

题,本文提出了一种高质量的体绘制算法。我们采用三维纹理映射作为核心的绘制算法,结合可见性测试、遮挡测试和模板测试来加快绘制速度。而且,利用体素计数表和滤过的可见子块,排除只有很少可见体素的子块,减少了在 CPU 和 GPU 之间的数据传输量。实验结果表明,对 $512 \times 512 \times 512$ 的体数据,可以在不损失图像质量的前提下,以可交互的速度进行绘制。

1 基于 GPUs 的体绘制算法

大部分基于显卡的体绘制依靠三维纹理的硬件。文献[4]利用显卡可以对 2 563 的体数据进行实时的绘制;文献[5]提出了一种基于多通道的绘制技术的基于显卡的体绘制算法。最近,利用可以编程的显卡,各种优化技术(如:提前不透明度截止^[6]和空体元空间跳跃技术^[7])已经有效运用到基于显卡的体绘制中。

本文算法的整个流程划分为四个步骤,如图 1 所示,其中空子块的空间跳跃和提前子块截止是以每个子块来进行的操作。

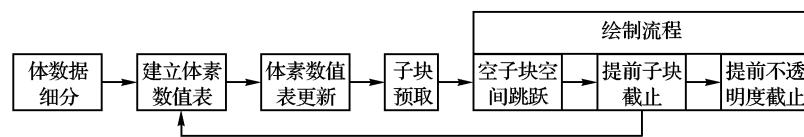


图 1 绘制流程

收稿日期:2006-11-08;修订日期:2007-01-12 基金项目:中法先进计划资助项目(PRA SI 03-03)

作者简介:谢凯(1974-),男,湖北潜江人,讲师,博士,主要研究方向:医学图像处理; 杨杰(1963-),男,教授,博士生导师,主要研究方向:图像处理、模式识别; 朱跃敏(1958-),男,教授,博士生导师,主要研究方向:医学图像处理。

作,而提前不透明度截止是以每个体素来进行的操作。

1.1 体素的计数表设计

对每一个子块,本文构造了一个体素计数表来统计经过透明传输函数分类后可见体素的数量。体素计数表被用来创建滤后可见子块,以有效地实现空体元的空间跳跃。对一个给定的子块体素计数表构造如下:

首先,我们创立一个一维的数组,由一个序列的体素计数器组成,它的索引与某个体素的灰度范围相对应。然后遍历子块中的所有体素,记录它们的灰度值,将其增加到相应的体素计数器中。最后,所有的体素计数器被综合得到一个体素计数表中。

一旦构造了一个体素计数表,对一个给定的透明传输函数,我们能够通过计算 R_1 和 R_2 ,得到一个子块中的可见体素比例。

$$\begin{cases} R_1 = \frac{VCT[OTF_{MAX}]}{VCT[VCT_{MAX}]} \\ R_2 = \frac{VCT[OTF_{MIN}]}{VCT[VCT_{MAX}]} \end{cases} \quad (1)$$

这里, OTF_{MAX} 和 OTF_{MIN} 表示一个透明传输函数中的最大和最小的灰度值, VCT_{MAX} 表示体素计数表的最大值。当所有体素中的灰度值大于 OTF_{MAX} ,公式(1) 中的 R_1 为 0。同样的,当所有体素中的灰度值小于 OTF_{MIN} , R_2 为 1。如果 R_2 为 0 或 R_2 为 1,则认为所指定的子块为空。

1.2 子块的可见性测试

对包含许多空体元的大体数据,我们需要根据透明传输函数分类的结果,确定哪一个子块是空区。而且,有的非空的子块只包含少量需要绘制的体素。利用体素计数表,连同用户定义的两个阈值 α 和 β ,对这些几乎为空的子块进行滤波。我们称通过滤波操作后仍然保留下来的子块为滤后可见子块。对每一个子块计算不受视点约束的可见性信息,构造滤后可见子块的步骤如下:

- 1) 利用用户定义的透明传输函数的最小值(OTF_{MIN}) 和最大值(OTF_{MAX}) 测试所有子块;
- 2) 对当前访问的子块,如果公式(1) 中的 R_1 大于 α 或 r_1 小于 β ,将该子块加入滤后可见子块;
- 3) 既然滤后可见子块依靠透明传输函数,当透明传输函数变化时,我们更新滤后可见子块。

1.3 子块的预取

通常,在绘制循环中取出子块。当视点改变时,需要从显存中释放一些子块,加入一些新的可见子块。显存始终处于被充分利用的状态,这样会引起严重的性能退化。为了解决这个问题,不管视点的改变和其他被遮挡的子块,本文预取高可能性被绘制的子块。边界子块被定义成滤后的可见子块的子集,它接近于围绕原始体数据的边界面。我们在绘制循环之前取来这些边界子块,并将显存分配给它们。在绘制循环中,预取的子块不被其他子块取代。

1.4 绘制循环

当一个大体数据被绘制时,许多部分(子块)被其他部分遮挡。而且,依靠给定的透明传输函数,存在许多空的部分。通过避免不必要的遮挡的和空的部分,我们能够提高总的绘制性能。为了解决这些问题,本文提供了不同级别的可见性测试,例如:逐个子块级别的空体元测试,逐个子块级别的遮挡测试,逐个体素级别的不透明测试,所有这些都是基于遮挡映射和显卡加速来执行的。在本文的算法中,只有通过 3 个

可见性测试的子块和未被遮挡的体素才考虑被绘制,可见性测试的详细描述如下:

空子块空间跳跃:空子块空间跳跃的目的是确定子块包含的体素透明度是否都为 0,如果是,则去掉这些子块。在本文的算法中,对每一个子块,核对它是否属于滤后可见子块,如果不是,则不对它进行绘制。

提前子块截止:既然将整个大体数据细分成许多大的子块,在某一视点下,一些子块完全被其他子块所遮挡。在这种情况下,我们不对该子块继续绘制流程中的其他操作,如非常费时的逐个像素操作:三维纹理映射和模板测试。作为解决办法,我们使用遮挡询问,看该子块的边界范围是不是完全被当前的遮挡映射所遮挡。更进一步地,在当前的遮挡映射下,我们可以绘制与轴向平行的子块包围盒,通过遮挡询问统计有多少体素可以真正的被绘制。如果体素的计数为 0,在后面的绘制中不考虑该子块。如果体素的计数为不为 0,将对该子块的体素进行提前不透明度截止的测试。

提前不透明度截止:当子块被绘制时,通过排除被已经绘制的部分所遮挡的体素来提高绘制速度。既然切片以从后到前的次序来绘制,在绘制子块时,不能得到它们遮挡信息。作为解决办法,遮挡信息可以从已经被绘制部分中获得。在每一个子块被绘制后使用模板缓存建立一个映射,在从后往前的绘制过程中,如果积聚的不透明度大于设定的阈值,则将其模板位置为 1,其后面的体素将不被绘制。

2 实验结果与分析

2.1 运用实例

既然三维纹理映射被大多数显卡所支持,允许纹理的大小为 2 的幂次方,本文需要将整个体数据分成小块的纹理。例如:我们可以将一个 $512 \times 512 \times 512 \times 2B$ 的体数据分成 512 个子块,每一个子块为 $64 \times 64 \times 64 \times 2B$ 。但是,一个过多的细分在三维纹理中会引起一个上下切换的问题。为了解决这个问题,本文在显存中建立了 8 个大小为 $256 \times 256 \times 256$ 三维纹理,当需要时从这 8 个纹理中的一个取出一个 $64 \times 64 \times 64$ 大小的子块,只需要 8 个上下切换(在最坏的情况下),这种技术能够减少上下切换的次数。

本文用 VC++ 实现我们的体绘制算法,用到了 Pixel Shader 2.0 和 DirectX 9.0,在配有 P4 2.4GHz CPU, NVIDIA Geforce 6600 显卡(128MB)的 PC 机上对本文的算法进行测试。实验中的基本的体数据模型如表 1 所示,大小分别为 213MB, 256MB 和 564MB。

2.2 性能分析

对基本模型本文算法的性能分析和可见性测试结果如图 2(a) 和图 2(b) 所示。对 256MB 的体数据,以 512×512 窗口进行绘制的速度为 3 ~ 4fps。在图 2(a) 中,将本文的算法与基于软件的直接体绘制算法^[8] 和基于硬件的没有可见性测试的体绘制算法进行了对比,HW, SW 和 OURS 分别表示基于硬件的没有可见性测试的体绘制算法,基于软件的直接体绘制算法和本文的体绘制算法。实验结果表明本文的算法在性能上优于其他算法。例如;在图 2(a),由于空体元的空间测试和提前不透明度截止,本文算法的绘制速度是基于软件的直接体绘制算法的 15 倍。而在图 2(b) 中,绘制的子块的数量,不同形状表示不同的体数据,由于空体元的空间跳跃测试和遮挡测试,许多子块被忽略。本文算法所绘制子块大小只占原始体数据大小的 15% ~ 30%。图 3 为利用人体 CT

扫描数据得到的体绘制结果图。数据大小分别是(a) $512 \times 512 \times 426 \times 2B$, (b) $512 \times 512 \times 512 \times 2B$, (c) $512 \times 304 \times 1730 \times 2B$, 绘制的窗口为 512×512 , 没有任何的图像质量损失。

表1 基本体数据(绘制结果见图3)

体数据	分辨率	切片数量	单个体素大小/B	体数据大小/MB
腹部	512×512	426	2	213
头部血管	512×512	512	2	256
虚拟人	555×308	1730	2	564

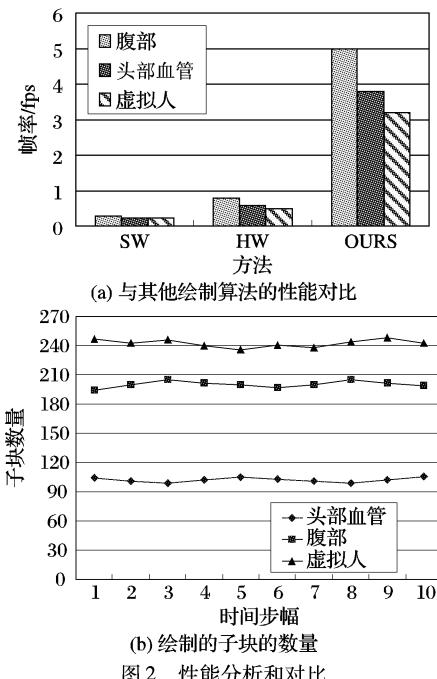


图2 性能分析和对比

3 结语

本文提出了一种基于显卡的硬件加速体绘制算法,能够对比显存容量大的体数据进行绘制。由于采用了可见性测试和遮挡测试的技术,加快了绘制速度,能够对大体数据以可交互的速度进行绘制。本算法也存在不足,主要是由于CPU和GPU之间的数据传输速度造成的,我们希望新一代的PCI-E总线技术能够解决这个问题。

(上接第1128页)

不需要等到跟踪到完整的目标轨迹。然而,当训练轨迹较长且数目众多时,深度优先搜索算法的时、空效率是一个需要考虑的问题。幸运的是在数据挖掘领域已经提出了许多快速的序列模式挖掘算法,如何将这些算法应用于目标运动轨迹的分析和理解将是下一步的研究方向。

参考文献:

- [1] COLLINS R. A system for video surveillance and monitoring: VSAM final report. TR: CMU-RI-TR-00-12[R]. Carnegie Mellon University, 2000.
- [2] HARITAOGLU I, HARWOOD D, DAVIS L. W4: Real-time surveillance of people and their activities[A]. IEEE Transaction on Pattern Analysis and Machine Intelligence[C]. 2000. 809–830.
- [3] YANG M, AHUJA N. Extraction and classification of visual motion pattern recognition[A]. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition[C]. 1998. 892–897.

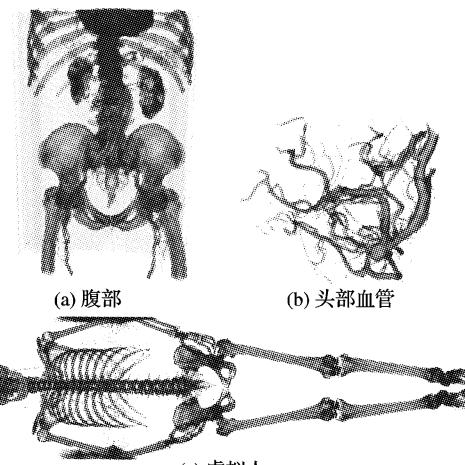


图3 体绘制结果

参考文献:

- [1] BAJAJ C, IHM I, PARK S. Compression-based 3D texture mapping for real-time rendering[J]. Graphical Models, 2000, 62(6): 391–410.
- [2] STRENGERT M, MAGALLON M, WEISKOPF D, et al. Large volume visualization of compressed time-dependent datasets on GPU clusters[J]. Parallel Computing, 2004, 31(2): 205–219.
- [3] POMMERT A, HÖHNE KH. Evaluation of Image Quality in Medical Volume Visualization: The State of the Art [A]. Medical Image Computing and Computer-Assisted Intervention. Proc, MICCAI 2002 Part II Lecture Notes in Computer Science 2489[C]. Springer Verlag Berlin, 2002. 598–605.
- [4] CULLIP TJ, NEUMANN U. Accelerating Volume Reconstruction with 3D Texture Mapping Hardware. TR93-027[R]. University of North Carolina, 1993.
- [5] MEIBNER M. Enabling Classification and Shading for 3D Texture Mapping based Volume Rendering using OpenGL and Extensions [A]. IEEE Visualization 99[C]. 1999. 207–214.
- [6] MATSUI M, INO F, HAGIHARA K. Parallel volume rendering with early ray termination for visualizing large-scale datasets[A]. ISPA 2004[C]. 2004. 245–256.
- [7] LI W, MUELLER K, KAUFMAN A. Empty space skipping and occlusion clipping for texture-based volume rendering[A]. IEEE Visualization 2003[C]. 2003. 317–324.
- [8] LEVOY M. Efficient ray tracing of volume data[J]. ACM Transactions on Graphics, 1990, 9(3): 245–261.

- [4] HU WM, XIE D, TAN TN. A hierarchical self-organizing approach for learning the patterns of motion trajectories[J]. IEEE Transaction on Neural Networks, 2004, 15(1): 135–144.
- [5] OWNES J, HUNTER A. Application of the self-organizing map to trajectory classification[A]. Proceedings of IEEE Workshop on Visual Surveillance[C]. 2000. 77–83.
- [6] KOHONEN T. Improved versions of learning vector quantization [A]. International Joint Conference On Neural Networks[C]. 1990. 545–550.
- [7] 潘梅森, 颜君彪. 基于图像块动态调整的码子内再匹配矢量量化[J]. 计算机应用, 2006, 26(3): 592–594.
- [8] 高新波. 模糊聚类分析及其应用[M]. 西安: 西安电子科技大学出版社, 2004. 50–53.
- [9] 吕静, 王晓峰, ADJEI O, 等. 序列模式图及其构造算法[J]. 计算机学报, 2004, 27(6): 784–788.
- [10] WU S, MANBER U. Fast text searching allowing errors[J]. Communication of The ACM, 1992, 35(10): 83–91.