医学图像大体数据快速体绘制算法研究

王阳萍[1] 党建武[1] 杜晓刚[1] 李 莎[2] 田种泽[2]
(兰州交通大学电子与信息工程学院 兰州 730070)[1] (兰州军区兰州总医院放疗科 兰州 730050)[2]

摘 要 为了在保证绘制质量的前提下有效提高大的医学体数据的绘制速度,提出了一种快速的体绘制算法。该算法将大的体数据分割成等大小的数据块,然后通过对每一数据块进行空白数据块的空间跳跃、提前数据块截止和提前光线截止的可见性测试来加快体绘制的速度,最后使用体绘制预积分来提高体绘制的图像质量。实验结果表明,对大的体数据,可以在不损失图像质量的前提下,实现快速的绘制。

关键词 医学图像,大体数据,体绘制,可见性测试,预积分
中图法分类号 TP317 **文献标识码** A

Research on Fast Volume Rendering of Medical Image with Large Volume Data

WANG Yang-ping[1] DANG Jian-wu[1] DU Xiao-gang[1] LI Sha[2] TIAN Zhong-ze[2]
(School of Electronic & Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China)[1]
(Department of Radiotherapy, Lanzhou General Hospital of Lanzhou Command, Lanzhou 730050, China)[2]

Abstract In order to do fast volume rendering for a large medical volume data and meanwhile assure the quality of rendering, a fast volume rendering method was presented. Firstly, the large medical volume data was divided into equal-size blocks, and then a set of visibility tests such as empty block space skipping, early block termination and early ray termination were used to speed up the whole rendering process. Finally, volume rendering pre-integration was utilized to improve the performance of volume rendering. The experiment results show that the proposed fast volume rendering has picked up the speed of rendering for a large medical volume data without loss of image quality.

Keywords Medical image, Large volume data, Volume rendering, Visibility test, Pre-integration

1 引言

医学体数据场的可视化就是运用计算机图形学的理论和方法,将由一系列二维CT切片组成的体数据集转换为三维图形,以更直观的方式揭示人体器官的空间分布关系。它不仅能够提供大量有用的人体信息,而且对改进外科手术计划和医疗教育有很重要的作用。因此,近十多年来,医学体数据场的可视化一直是计算机图形学的一个研究热点[1-3]。

为了取得更好的绘制效果,从中获得更多有用信息,并满足医生及病人对三维体的实时观察及旋转等操作,需要实现实时快速的三维绘制。然而,随着医学影像分辨率的提高,医学影像采集设备产生了庞大的二维源数据,如一个CT扫描仪器可以产生多达1000多张切片的二维医学图像,待处理的数据量可达上千兆。虽然现在的显卡技术可以使体绘制技术达到可交互的速度,但是显卡的有限显存却不能处理大的体数据。因此,为大的数据体提供一种快速的三维体绘制方法是一项很有挑战性的任务[4]。

为了对医学图像大体数据进行实时体绘制,本文提出了一种快速的体绘制算法。该算法先将大数据体分割成小的数据块,然后利用可见性测试(空数据块的空间跳跃、提前数据块截止和提前光线截止)逐一处理这些数据块,进而减少数据量。最后,使用体绘制的预积分来进一步改善体绘制方法的执行效率和图像质量。实验结果表明,对大的医学体数据,不仅可以获得很好的图像质量,而且可以实现快速的绘制。

2 快速体绘制算法流程

本文提出的体绘制算法,整个处理流程分为5个步骤,如图1所示。其中空数据块的空间跳跃和提前数据块截止是以数据块为单位进行操作的,而提前光线截止是针对每个体素进行处理。

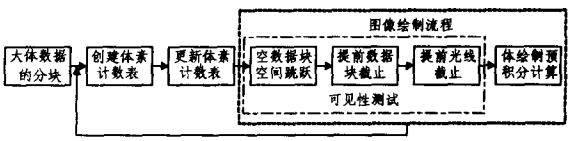

图1 快速体绘制算法流程图

3 算法分析与设计

3.1 体数据分块

在进行体绘制之前,首先将大的体数据集分割成相同大小的数据块[5,6],并按顺序将数据块逐一放入缓存中处理,这

到稿日期:2010-01-08 返修日期:2010-04-01 本文受国家863高技术研究发展计划基金项目(2006AA02Z499),国家自然基金项目(60962004),甘肃省科技攻关计划项目(0708GKCA047)和甘肃省自然科学基金项目(0803RJZA015)资助。

王阳萍(1973-),女,副教授,主要研究方向为医学图像处理,E-mail:wangyangping@mail.lzjtu.cn;党建武(1963-),男,教授,博士生导师,主要研究方向为智能信息处理;杜晓刚(1985-),男,硕士生,主要研究方向为数字图像处理。

样不仅提高了缓存的利用率，而且提高了数据处理速度。

3.2 创建体素计数表

为了能剔除不可见的数据块，有效地实现空白数据块的空间跳跃，需要统计每一数据块中的可见体素数。通过对每个数据块进行不透明度转换函数的分类来获得相应数据块中包含的可见体素数，并将这一数值保存到体素计数表中。对一个给定的数据块体素计数表构造如下：首先，创建一个一维的数组，由一系列的体素计数器组成，它的索引与某个体素的灰度值范围相对应；然后遍历数据块中的所有体素，记录它们的灰度值，并根据其灰度值范围，增加到相应的体素计数器中；最后，将所有的体素计数器都综合到一个体素计数表中。

在构建好数据块的体素计数表后，对于给定的不透明度转化函数，通过计算 R_1 和 R_2 来判断这个数据块是否为空：

$$\begin{cases} R_1 = VCT_{Min}/OTF_{Max} \\ R_2 = VCT_{Max}/OTF_{Min} \end{cases} \tag{1}$$

式中，OTF_{Max} 和 OTF_{Min} 分别表示该不透明度转化函数中的最大灰度值和最小灰度值，VCT_{Max} 和 VCT_{Min} 分别表示体素计数表中的灰度最大值和最小值。当体素计数表的最小值 VCT_{Min} 大于 OTF_{Max} 时，式(1)中的 R_1 为 1；而当体素计数表的最大值 VCT_{Max} 小于 OTF_{Min} 时，式(1)中的 R_2 为 0。如果 R_1 为 1 或者 R_2 为 0，则认为所指定的数据块为空。

3.3 更新体素计数表

在绘制包含了很多数据块的大体数据集之前，需要利用不透明度转换函数将这些数据块中的可见部分分离出来。但是，在这些被分离出来的可见数据块中，有的数据块中只包含了少量的可绘制体素。为了提高绘制速度，定义了一个基于可见数据块的可见体素比率等式：

$$VVR = \frac{VVT[OTF_{Min} \sim OTF_{Max}]}{AVB} \tag{2}$$

式中，$VVT[OTF_{Min} \sim OTF_{Max}]$ 是根据体素计数表获得的一数据块中的所有可见体素的数目，AVB 是该数据块中所有体素的数目，VVR 代表数据块的可见体素比率。

利用这个可见数据块的可见体素比率等式和一个由用户自定义的门限值 α，算法进一步选择可见数据块的步骤如下：

(1)对每一可见数据块，计算其数据块的可见体素比率。

(2)对当前访问的可见数据块，如果其可见体素比率小于门限 α，就将这一数据块标识为不可见，然后更新原先的体素计数表。如果其可见体素比率大于门限 α，则执行下一步处理。

(3)因为这些可见数据块依赖于不透明转化函数，所以，当不透明转化函数发生变化后，要更新可见数据块。

(4)使用一个由用户自定义的门限来过滤这个包含少量可见体素的可见数据块，那些通过了过滤操作仍存在的数据块叫做可见数据块。

(5)根据最终剩下的可见数据块，更新原先的体素计数表。

3.4 数据块的可见性测试

当绘制一个大体数据时，根据给定的不透明转化函数，会产生很多空的部分。而且，许多部分将会被其它的部分遮挡。因此，通过避免不必要的空白部分和遮挡，可以提高总体的绘制性能。为了解决这些问题，本文研究了一系列的可见性测试[7,8]。在本文的算法中，只有通过以下可见性测试的数据块和未被遮挡的体素才能绘制大体数据。具体方法如下：

(a) 空白数据块的空间跳跃：空白数据块空间跳跃的目的是确定测试数据块所包含的体素透明度是否都为 0。如果是，则去掉这些数据块。在本文的算法中，针对每一测试数据块判定它是否是可见的数据块。如果不是可见数据块，那么直接跳过该数据块；如果是可见的数据块，则对其进行提前数据块测试。

(b) 提前数据块截止：由于大的数据体被分割成了很多小的数据块，在某个视点下，一些数据块将完全被其它的数据块所遮挡。在这种情况下，为了进一步减少数据运算量，不应该对这种数据块继续进行其它绘制操作。因此，算法中为数据块绘制一个与轴向平行的方向包围盒，利用遮挡剔除方法，不仅可以判断数据块的包围盒边界是否完全被当前的遮挡映射所遮挡，而且可以统计出有多少体素可以真正地被绘制。如果统计的体素数为 0，那么在后面的绘制中将不再考虑该数据块；如果统计的体素数不为 0，那么该数据块就可以进入提前光线截止的测试。

(c) 提前光线截止：对剩下的可见数据块，光线按照由前到后的顺序对数据块进行取样。在取样过程中，光线上累加的取样点不透明度如果大于某一给定的阈值，则停止在该光线方向上对其它数据块的任何取样操作。

3.5 体绘制预积分

由于体绘制的颜色和不透明度信息都是在不连续的取样点上获得的，为了取得更好的绘制效果，往往需要设置更多的取样点，并对这些取样点进行更精确的插值计算。但这样就使得体绘制的运算量变大，绘制速度变慢。因此，为了能进行快速、高质量的体绘制，本文使用了体绘制的预积分方法[9]。利用体绘制的预积分计算，可以获得与经过了所有可见性测试的数据块相关的预积分查找表。该查找表不仅保存了所有取样点的颜色和透明度信息，而且保存了两取样点之间的所有颜色和透明度信息。

体绘制预积分查找表的生成分为两部分：首先沿着视线方向对可见数据块进行采样，接着对两两相邻的取样点进行体绘制积分，并将计算结果保存到预积分查找表中。然而，这样的预积分查找表的生成往往比较费时。为了能减少计算时间，本文采用加速的预积分计算方法：

(1)在沿着视线方向对可见数据块进行采样时，使用固定的、等大小的取样间距。

(2)在转换函数只对某一特定的标量值做局部变动时，无须重新计算整个查找表，而只对产生变化的那一部分进行重新计算。

(3)在用颜色转换函数 $c(s)$ 和不透明度转化函数 $o(s)$ 计算标量值 s 的颜色和不透明度时，利用积分函数来实现对其颜色和不透明度的加速求解。这样，原先的颜色转化函数就近似变成：

$$C(s_0, s_1, d) \approx \frac{d}{s_1 - s_0}(K(s_1) - K(s_0)) \tag{3}$$

式中，s_0，s_1 分别代表两个不同取样点的标量值，d 为取样间距。$K(s)$ 是对颜色转化函数进行积分的函数：$K(s) = \int_0^s c(s)\mathrm{d}s$。

原先的不透明度转换函数就近似为：

$$O(s_0, s_1, d) \approx 1 - \exp\left(\frac{d}{s_1 - s_0}(T(s_1) - T(s_0))\right) \tag{4}$$

式中，s_0，s_1 和 d 分别代表两个不同取样点的标量值和取样间距。$T(s)$ 是对不透明度转换函数进行积分的函数 $T(s) = \int_0^s o(s)\mathrm{d}s$。

(下转第 279 页)

[14] Zheng Shijue, Zhang Yan, He Tingting. The Application of Genetic Algorithm in Embedded System Hardware-software Partitioning[C]// International Conference on Electronic Computer Technology: 219-222
[15] 高健，李涛. 三种软硬件划分算法的比较分析[J]. 计算机工程与设计，2007，29(7)：3426-3428
[16] 李涛，杨愚鲁，马平，等. 基于遗传算法的可重构系统软硬件划分[J]. 计算机工程与应用，2007，26(3)：56-58
[17] 范乐君，李斌，庄镇泉，等. 一种基于 DQCGA 算法的软硬件动态划分方法[J]. 计算机科学，2008，35(5)：201-205
[18] Wu J G, Srikanthan T, Zou G W. New model and algorithm for hardware/software partitioning[J]. Joural of Computer Science and Technology, 2008, 23(4): 644-651
[19] Yuan Mingxuan, He Xiuqiang, Gu Zonghua. Hardware/Software Partitioning and Static Task Scheduling on Runtime Reconfigurable FPGAs Using a SMT Solver[C]// IEEE Real-Time and Embedded Technology and Applications Symposium. 2008: 295-304

(上接第 251 页)

4 实验结果与分析

根据以上给出的算法，可以将一个大小为 512×512×512×2B 的体数据分割成 64 块，每一数据块的大小为 128×128×128×2B。但是，对体数据进行的分割过多，就会引起一个上下文切换的问题。为了解决这个问题，实验中将体数据分割成了 8 个大小为 256×256×256×2B 的数据块。这样，不仅可以有效地利用显存，而且可以减少上下文切换的次数。

本文采用 VC++6.0 实现了快速的体绘制算法，在配有 Pentium(R) Dual E2140 CPU、1G 内存、独立显卡(128MB)的 PC 机上对本文提出的算法进行测试。实验中的体数据模型如表 1 所列，大小分别为 84MB，176MB，253MB 和 428MB。

表 1 测试体数据集

体数据集	分辨率	切片数目	单个体素大小/B	体数据集大小/MB
心脏	512×512	168	2	84
脚部	512×512	352	2	176
头部	512×512	506	2	253
人体上半部分	512×512	856	2	428

在对大的体数据集进行体绘制时，本文所提出的算法的性能分析结果如图 2 所示。对于 428 MB 的大数据体，以 400×400 窗口进行绘制的速度为 8～11fps。在图 2 中，利用相同的 PC 机和相同的测试体数据集，将本文所提出的算法在不使用任何图像加速硬件的情况下同传统的直接体绘制算法进行了比较，DVR，OURS 分别表示传统的绘制算法和本文所提出的算法。实验结果表明，本文的算法在性能上比传统的直接体绘制算法要好。由于本算法使用了可见性测试，大量的空白数据和对最终图像不做任何贡献的数据被剔除，因此其速度约为传统直接体绘制算法的 5～7 倍。图 3 是对人体 CT 体数据集进行快速体绘制的结果图。各体数据集的大小分别是(a) 512×512×168，(b) 512×512×352，(c) 512×512×506，(d)512×512×856，绘制窗口大小为 400×400。从图 3 中可以看到，在使用了体绘制的预积分后，绘制图像没有任何质量损失。

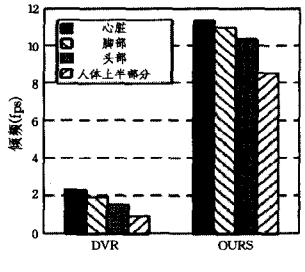

图 2 体绘制算法的性能比较

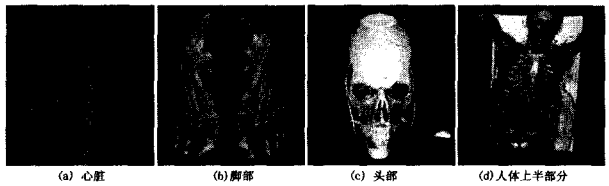

图 3 快速体绘制结果

结束语 随着现代医学影像技术的发展，所获得的医学影像体数据集的数据量也越来越大。因此，如何对大的体数据集进行快速、高质量的体绘制，是一个很大的挑战。鉴于此，本文提出了一种快速体绘制算法。通过将大的医学体数据分割成小的数据块来减少对显存的需求；对这些数据块进行一系列的可见性测试，将全部的体数据块提炼为一个近似的可见集，从而在很大程度上减少了体绘制要处理的数据量，实现了对大体数据的快速体绘制；利用体绘制预积分技术，提高了体绘制的图像质量。但是，在转换函数发生改变时，我们需要重新计算整个体数据集，可实际上，在上一转换函数中所产生的数据仍可被新的转换函数使用。因此，如何有效地利用转换函数所产生的数据，将是一个重要的研究课题。

参 考 文 献

[1] 段宝山，潘振宽. 医学断层图像三维重建的辅助轮廓线法[J]. 计算机辅助设计与图形学学报，2005，17(8)：1862-1866
[2] 鲍苏苏，林斌. 医学图像三维表面重建分叉曲面的数学形态学研究[J]. 计算机科学，2003，30(12)：136-138
[3] 何青，刘允才. 股骨模型的三维重建与变形[J]. 计算机工程，2007，33(13)：221-223
[4] Levoy M. Display of surfaces from volume data[J]. IEEE Computer Graphics and Applications, 1988, 8(3): 29-37
[5] Kye Heewon, Jeong Dong kyun. Accelerated MIP based on GPU using block clipping and occlusion query[J]. Computers and Graphics, 2008, 32(3): 283-292
[6] Choi Jae-Jeong, Shinb Byeong-Seok, Shina Byeong-Seok, et al. Efficient volumetric ray casting for isosurface rendering[J]. Computers and Graphics, 2000, 24(5): 661-670
[7] Levoy M. Efficient ray tracing of volume data [J]. ACM Transactions on Graphics, 1990, 9(3): 245-261
[8] Levoy M. Volume rendering by adaptive refinement [J]. The Visual Computer, 1990, 6(1): 2-7
[9] Lum E B, Wilson B, Ma K-L. High-quality lighting and efficient pre-integration for volume rendering[J]. Euro Graphics/IEEE Symposium on Visualization, 2004: 25-34