

基于光线相关性的快速光线投射算法

胡英^{1),2)} 徐心和¹⁾

¹⁾(东北大学人工智能与机器人研究所,沈阳 110004) ²⁾(沈阳东软数字医疗系统股份有限公司,沈阳 110179)

摘要 光线投射算法是一种应用广泛的体绘制技术的基本算法,其存在的主要问题是绘制速度较慢。为了提高光线投射算法的绘制速度,以满足医学图像三维重建的应用需求,在深入研究和比较各种光线投射加速算法的基础上,提出了以接近云算法为核心的、适用于医学图像三维重建的综合性加速算法,并在PC机平台上实现了该算法,在保证图像质量的同时绘制速度提高了一个数量级左右,为医学图像三维重建的实用化提供了有效的手段。

关键词 医学图像 光线投射 光线相关性 体绘制 接近云

中图分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2004)02-0234-06

Accelerate Ray Casting Algorithm Base on Ray Coherence

HU Ying^{1),2)}, XU Xin-he¹⁾

¹⁾(Institute of AI and Robotics, Northeastern University, Shenyang 110004)

²⁾(Shenyang Neusoft Digital Medical Systems Co., Ltd, Shenyang 110179)

Abstract Medical image 3D reconstruct is an important application filed for volume rendering, for it special using, it required fast interactive speed and high image quality. Ray casting is a widely used basic volume rendering algorithm. It can get high quality image but the rendering speed is very slowly. In order to improve its rendering speed and apply it to medical image 3D reconstruct, a synthesized accelerate algorithm is proposed in this paper. Firstly, accelerate algorithms for ray casting are fully studied and compared. Secondly, proximity clouds algorithm has been selected and extended to continuous ray casting. Then the synthesized accelerate algorithm base on ray coherence has been proposed. Finally, the algorithm is realized on PC platform. Four data sets are used for test at two different image resolutions. The result shown that compare with standard ray casting algorithm, the accelerate algorithm not only improve the rendering speed for nearly ten times but also bring no depress on image quality. At 512×512 image resolution, a nearly real time interactive speed has been get. It gives a more capability method for the application of medical image 3D reconstruct.

Keywords medical image, ray casting, ray coherence, volume rendering, proximity clouds

引言

光线投射算法是体绘制技术的基本算法之一。由于它具有原理简单、易于实现、可以生成高质量的显示图像等优点,因而得到了广泛的研究和应用。与基于传统计算机图形学技术的绘制方法相比较,体绘制技术存在的主要问题就是难以利用现有的图形显示硬件设备进行加速,因此如何提高绘制速度是体绘制技术中需重点研究的问题。

目前,针对于光线投射算法的加速技术主要有空间剖分(space subdivision)技术和光线相关性(ray coherence)技术两种^[1]。基于空间剖分技术的加速算法,例如BSP算法、Octree算法、Kd-tree算法等,是从传统的图形学领域延伸过来的。它是利用数据空间的相关性,通过对数据空间进行剖分,将连续的空体元划分到一定的包围盒中,当光线在数据空间穿行时,碰到这些包围盒只需进行简单的求交运算,就可以略过整个空体元的包围盒,从而减少光线投射的步数,达到降低计算时间的目的。与基本的

光线投射算法相比,基于空间剖分的加速技术的光线投射算法需要增加树搜索和包围盒求交运算,当非空体元在数据空间的分布比较复杂时,空间剖分所产生的包围盒的数量将迅速增加,由此而导致的预处理时间的增加以及大量的树搜索和包围盒求交运算将极大地降低算法的效率。

基于光线相关性的加速技术是人们在通过对光线投射算法的特点进行深入研究的基础上提出的。它主要是通过挖掘光线在体数据空间穿行过程的一些相关性的特点,来减少光线投射的数量和穿行的步数,从而实现算法的加速。本文将重点讨论基于光线相关性加速技术的各种算法,并且从中总结并实现了适合于医学图像三维重建的综合加速技术。本文所讨论的算法都是针对连续的光线投射的,而不是离散的光线投射。

2 基于光线相关性的加速算法

2.1 算法介绍

基于光线相关性的加速算法主要基于以下一种或几种原理^[2]: 像空间相关性 (pixel-space

coherency)、物空间相关性 (object-space coherence)、光线间相关性 (inter-ray coherence)、空间跳跃 (space-leaping)、序列间相关性 (sequence coherence)。

(1) 像空间相关性 基于像空间相关性加速算法的核心思想是,图像平面的像素之间存在着很高的相关性,两个相邻的像素具有相似的,或者完全相同的颜色和深度。利用这种相关性,可以采用有间隔的方式发射光线,对于不发射光线的像素采用一定的方式来插值计算,从而达到算法加速的目的。最有代表性的有自适应重采样算法^[3],即从屏幕的行和列采取一定的间隔发射光线,计算它们的颜色,如果相邻的两个发射光线的像素之间的颜色发生较大变化,则在它们之间增加光线发射的数量,以避免显示结果的走样,否则采用插值的方式计算不发射光线的像素的颜色;用于面显示的加速算法^[4],即首先按照一定的间隔从屏幕像素发射光线,计算出相应的深度值,如图1(a)所示。对于两个发射光线像素之间的像素,按照距离最近的原则,从相邻发射光线的像素的深度出发,搜索分界点,从而确定该像素点的深度值,如图1(b)、图1(c)所示。

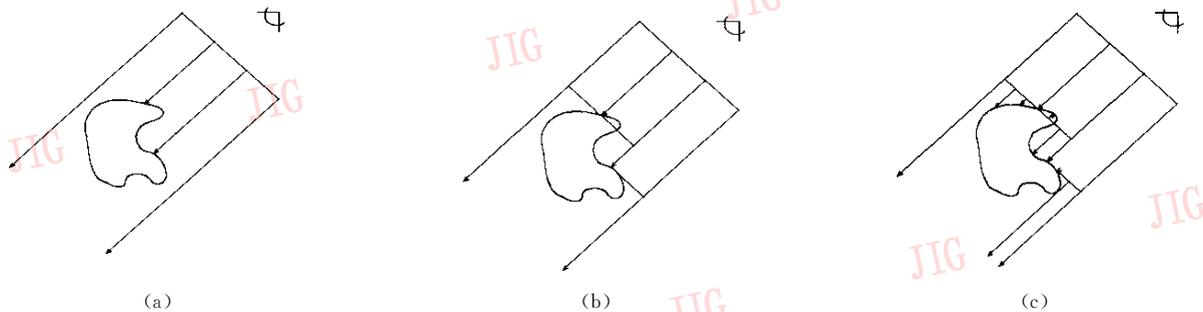


图1 一种基于像空间相关性的加速算法

(2) 物空间相关性 基于物空间相关性加速算法的核心思想是,数据空间的体素之间存在着很高的相关性,可以通过避免或减少在具有相同或者相近的物质区域内进行重采样的次数来实现算法的加速。具代表性的是自适应调整采样频率的方法^[5],即先进行低频采样(大步长采样),若两个采样点之间的值发生剧烈变化,则再进行密集的小步长采样。

(3) 光线间相关性 光线间相关性加速算法的核心思想是,在平行投影条件下,对于所有的光线,虽然它们的起点不同,但它们具有相同的斜率,当光线在体数据空间穿行时,具有相同的位移变化。Yagel 等人充分挖掘了光线间的相关性,提出了基于模板的光线投射算法^[6]。如图2所示,在连续的光

线投射中,由于同一步数上相邻的光线在体数据中的相对位置相同(如图中带圈的采样点),因此,可以记录一条光线穿行时在各步处的插值参数,其他光线穿行时不需要进行费时的三线性插值运算,只需要带入相应的参数即可。

(4) 空间跳跃 当光线在体数据空间穿行时有两种状态,一种状态是在空体素(透明体素)中穿行以搜索非空体素(不透明体素),另一种状态是在非空体素中进行颜色和阻光度的累积。由于在第1种状态下,空体素对于最终的显示结果没有任何影响,因此尽可能快地跳过空体素,将显著地加快算法的速度同时对图像的质量不会带来任何影响。实质上这也是基于空间剖分技术加速算法的核心思想。

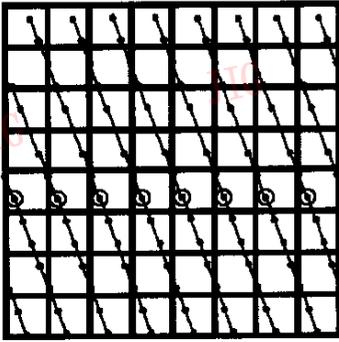


图 2 基于模板的光线投射算法

proximity clouds 算法^[7,8]是空间跳跃技术的一种非常优秀的实现方式。如图 3 所示,对于每个透明体素,记录它到距离它最近的不透明体素的距离,这样光线投射的时候,就可以以这个距离作为前进的步长,在这个距离范围内,沿任何方向前进,光线都不可能碰到不透明的体素。该方法极大地减少了光线投射的步数,大幅度提高了算法的效率。

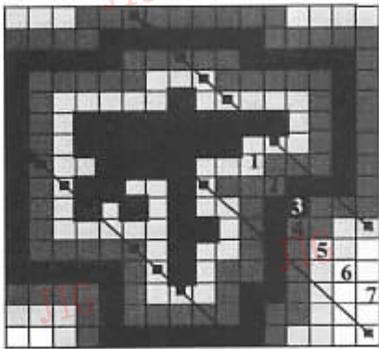


图 3 proximity clouds 算法

(5) 序列间相关性 当操作者进行旋转缩放等交互时,连续生成的图像之间存在着相关性,利用前一副图像中的有用信息可以有效地加快当前图像的渲染速度。针对这一特点,人们提出了相应的加速算法。如 coordinate-buffer(简称 C-buffer)方法^[9],首先将上一幅图像中屏幕各像素点发出的光线接触到的体数据空间的第 1 个非空体素的三维坐标,保存在一个内存缓冲区中(C-buffer),当进行旋转操作时,首先将 C-buffer 中保存的非空体素映射到新的图像屏幕上,那些有映射点的像素所发射的光线在体数据空间的穿行可以直接从它所对应的这个非空体素开始,只有那些没有映射点的像素才需要从起点开始穿行。Brady 等人将 C-buffer 方法进行了深入地推广,并引申到透视投影的成像方式中^[10]。Chen 等人进一步发掘了序列间的相关性^[11],提出

了一种关键帧的方法。即首先在主要的视线方向生成一系列的关键帧图像,然后,在进行旋转交互时,将与当前视线方向最接近的那一个关键帧图像映射到当前屏幕上,只从那些没有映射的像素投射光线。这种方法具有非常高的实时性,但是需要较长的预处理时间和较大的内存。

2.2 算法比较和讨论

基于像空间相关性和基于物空间相关性的加速算法在实现加速的同时都是以牺牲一定程度的图像质量为代价的。一般情况下,基于光线间相关性的加速算法要求光线之间的间距为一个体素的宽度,且体素必须是正方体,其不适用于透视投影的成像方式。很多体绘制技术的先驱例如 Yagel^[2],Levoy 都指出,利用数据空间的相关性略过空体素是提高速度的最根本手段。基于空间跳跃技术的 proximity clouds 算法正是这样一种加速算法,即首先,它尽可能地提高了光线略过空体素的效率,一是光线在空体素中以尽可能大的步长前进,同时还不需要进行费时的三线性插值计算;二是由于没有错过任何非空体素,所以不会对图像的质量产生任何影响。由于医学图像三维重建的目的是辅助医生进行诊断,它要求重建结果尽可能的准确,生成的图像质量尽可能的高,因此 proximity clouds 算法无疑是一种非常好的选择。由于用户需要进行大量的缩放和旋转等交互,因此利用序列间的相关性,加快交互时的图像绘制速度,将进一步提高系统的交互性能。

3 综合性加速算法的实现

综合性的加速算法,是基于空间跳跃技术的 proximity clouds 算法结合序列相关性的算法。

proximity clouds 算法的实现包括距离场的建立和光线的投射过程两个步骤。在 proximity clouds 算法中所建立的距离场为街区距离场^[7,8],街区距离的表达如下

$$D_p(\Delta x, \Delta y, \Delta z) = |\Delta x| + |\Delta y| + |\Delta z| \quad (1)$$

将所有非空体素的距离值设为 0,所有空体素的初始距离值设为 $+\infty$,则所有的空体素的距离值为

$$D_{i,j,k} = \min(D_{i,j,k}, D_p(o, p, q) + D_{i+o, j+p, k+q}) \quad (2)$$

$$o, p, q = 0, \pm 1$$

其中, $D_{i,j,k}$ 是坐标为 (i, j, k) 的空体素当前的距离值。由于距离场是与视线无关的,距离场的计算是在数据预处理的时候完成的。

对于任何一个空体素 $v(x, y, z)$, 设它的距离值为 D_v , 当前光线向量为 $I=(x_1, y_1, z_1)$ 。根据街区距离的定义, 沿着光线的方向进一步所行进的距离为

$$d = |x_1| + |y_1| + |z_1| \quad (3)$$

则光线在该空体素处可以行进的步数为

$$s_{\text{count}} = D_v/d = \frac{D_v}{|x_1| + |y_1| + |z_1|} \quad (4)$$

因此光线从该体素处行进所到达的下一个体素 $v(x', y', z')$ 的坐标为

$$x' = x + x_1 * s_{\text{count}} = x + \frac{x_1}{|x_1| + |y_1| + |z_1|} D_v$$

$$y' = y + y_1 * s_{\text{count}} = y + \frac{y_1}{|x_1| + |y_1| + |z_1|} D_v \quad (5)$$

$$z' = z + z_1 * s_{\text{count}} = z + \frac{z_1}{|x_1| + |y_1| + |z_1|} D_v$$

$$\text{令 } c_x = \frac{x_1}{|x_1| + |y_1| + |z_1|}$$

$$c_y = \frac{y_1}{|x_1| + |y_1| + |z_1|}$$

$$c_z = \frac{z_1}{|x_1| + |y_1| + |z_1|}$$

则光线在任意空体素 v 处的增量可表示为

$$\Delta_i = c_i \times D_v \quad i = x, y, z \quad (6)$$

这样光线投射的过程可以描述如下:

从屏幕像素发出一根光线进入体数据空间, 如果遇到空体素, 则根据式(6)计算光线的增量, 并确定光线所到达的下一个体素的坐标。光线行进到下一体素, 如果是空体素则重复前面的过程, 否则采用小步长在非空体素中进行重采样, 累计颜色和阻光度。直到光线穿透数据空间, 或者阻光度大于给定的阈值。

Proximity Clouds 算法是针对离散的光线投射而提出的。从前文的论述中可以看到, 在所建立的距离场中, 只有各体素处含有距离信息, 而对于连续的光线投射算法, 采样点并不总在各体素上, 大部分情况下都位于相邻的 8 个体素中间。在这种情况下, 如何确定各采样点处的距离值呢? 设采样点 S 与距离它最近的体素 V 的三维坐标差分别为 dx, dy, dz , ($dx, dy, dz < 0.5$), 则其街区距离为

$$D_p(S, V) = |dx| + |dy| + |dz| < 1.5 \quad (7)$$

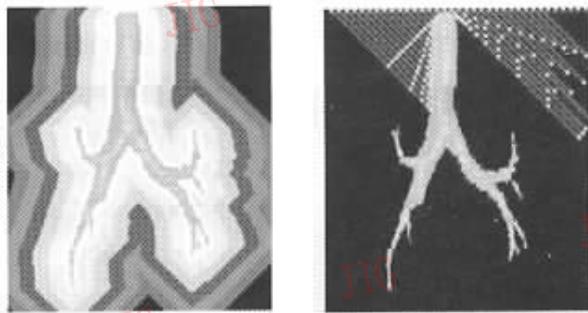
根据距离场的定义, 采样点 S 与体素 V 距离值的差距小于等于它们之间的街区距离, 即

$$|D_v(S) - D_v(V)| \leq D_p(S, V) \quad (8)$$

因此将采样点 S 的距离值取为 $D_v(V) - 1.5$ 显然是

非常安全的, 光线在采样点处以这个距离前进是绝对不会跳过非空体素的。这样就可以把 proximity clouds 算法应用到连续的光线投射中了。

下面以一个二维的例子来进一步说明 proximity clouds 算法。如图 4 所示, 图 4(a) 为一个组织的二维距离场示意图, 颜色从浅到深表示距离由小到大。图 4(b) 为从位于图像顶端的屏幕沿着某方向投射光线的过程, 浅色亮点为采样点。在这两个例子中采用 proximity clouds 加速算法的总的采样点和采用标准的 Ray casting 算法的总采样点分别为 168 和 10 398, 比例关系为 1 : 62。可见 proximity clouds 加速算法极大地减少了总的采样步数, 而且空体素比例越高, 加速效果越明显。



(a) 距离场示意图

(b) 光线投射示意图

图 4 Proximity Clouds 算法的二维示例

在用户进行旋转和缩放交互的时候, 采用渐进细化(progress refine)技术, 提高交互速度是一种非常通用的做法。在交互的时候, 从屏幕的行和列间隔一个像素投射光线, 生成的图像通过线性放大的方式显示出来, 这样在保证原分辨率的基础上, 将绘制时间降低为标准绘制的四分之一, 由于采用与标准绘制相同的采样步长和 shading 方法, 交互操作时的图像效果不会与标准绘制效果有太大的差异。另外, 当用户选定了感兴趣区域(ROI)停止交互进行观察时, 当前帧的视线方向与前一帧完全相同, 利用帧间的相关性, 可以将前一帧投射光线的像素颜色直接复制到当前帧, 只需要在屏幕上剩余的四分之三像素投射光线即可得到精确的最终显示图像, 从而进一步加快了图像绘制的速度。

4 试 验

结合以上的研究结果, 设计并实现了基于光线相关性的综合快速光线投射算法, 其系统结构如

图 5 所示。

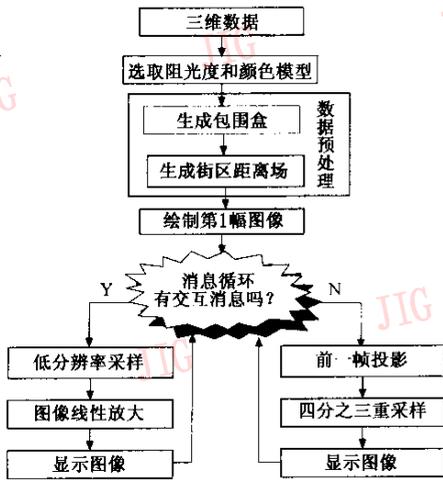
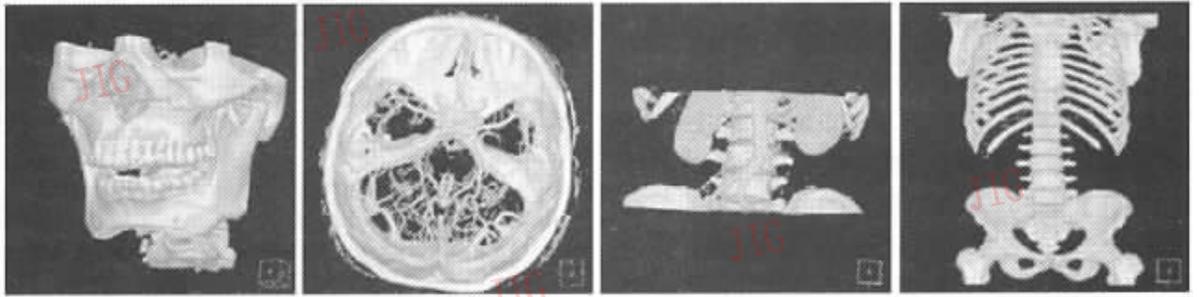


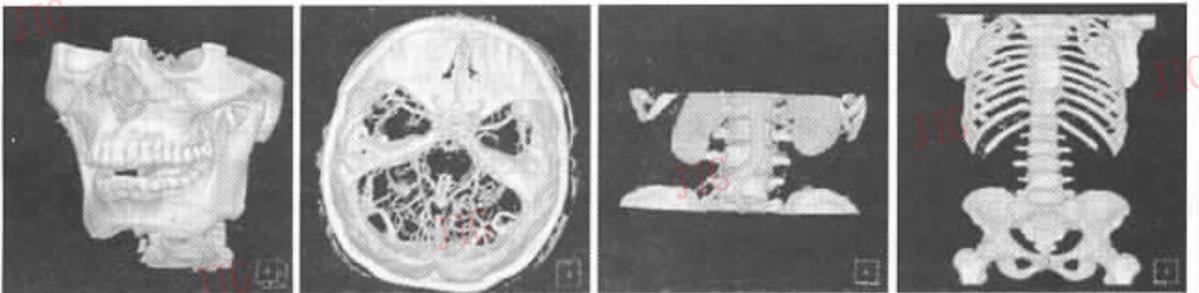
图 5 系统结构图

计算机配置为 Petium IV 1.7G, 512M DDR 内存。实验数据全部为 16bit 短整形, 采用三线性插值重采样和 phone shading, 屏幕分辨率(图像大小为 512×512), 标准的光线投射也应用了包围盒等常规加速技术。绘制结果如图 6 所示, 可以看出两种绘制算法的绘制结果完全相同。为测试加速性能, 绕数据场的 X 轴和 Y 轴旋转, 每 10° 取一个观察角度总共 72 个观察角度, 对两种算法的绘制时间进行了统计比较, 另外, 为了进一步验证算法的加速性能, 对于同样的实验数据, 在 1024×1024 图像分辨率下重复进行了实验。结果如表 1 所示。

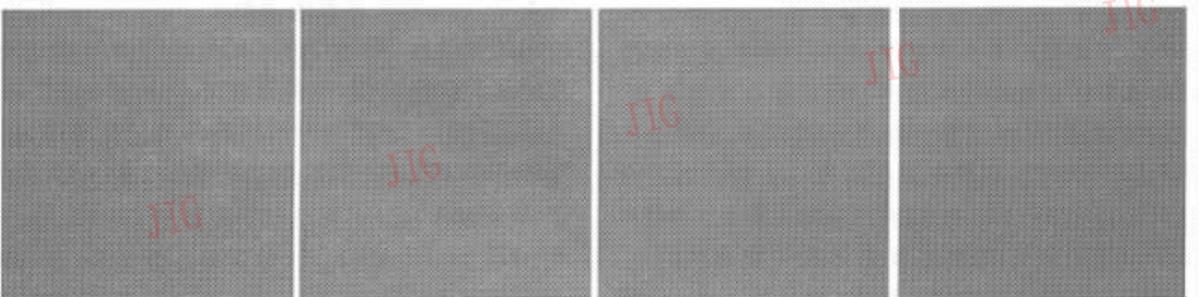
由于光线投射算法的特点, 不同视角图像的绘制时间是不同的, 最大和最小绘制时间是从不同的视角观察时, 图像绘制时间的最大值和最小值, 平均绘制时间是各观察角度绘制时间的平均值。



(a) 标准光线投射算法的绘制结果



(b) 本文加速算法的绘制结果



(c) 两种算法绘制结果之差

图 6 4 组实验数据的绘制结果

表 1 4 组实验数据加速前后的绘制时间

部位	数据场规模	图像分辨率	标准光线投射算法绘制时间(ms)			本文加速算法绘制时间(ms)			加速倍数	
			最大	最小	平均	预处理	最大	最小		平均
颅骨	512×512	512×512	10 331	7 634	8 500	2 824	1 032	484	672	12.65
	×115	1 024×1 024	73 172	39 007	46 484	2 824	4 844	2 011	3 078	15.10
头部 (血管)	512×512	512×512	12 401	1 703	6 679	3 531	1 216	281	703	9.50
	×156	1 024×1 024	55 938	13 531	27 110	3 531	9 017	1 063	3 508	7.73
腹部 (局部)	512×512	512×512	10 172	5 578	6 615	2 141	1 062	271	716	9.24
	×100	1 024×1 024	68 531	17 890	34 260	2 141	7 609	1 171	3 082	11.12
体部 (骨骼)	512×512	512×512	29 078	16 872	24 125	3 610	1 312	891	1 108	21.77
	×206	1 024×1 024	186 344	59 941	132 770	3 610	6 853	2 217	4 883	27.19

从实验结果来看,采用加速算法后,绘制速度都得到 10 倍左右的提高,在 512×512 的图像分辨率下,平均绘制时间均在 1s 左右。由于交互操作时的绘制时间是标准绘制的四分之一,交互速度可以达到每秒 4~5 帧,基本上可以满足用户实时交互的需求。另外,由于本加速算法只是跳过了对最终显示结果没有任何贡献的空体素,而对于光线投射过程中的非空体素均进行了三线性插值重采样,因此绘制结果与采用同样重采样算法的标准光线投射算法的绘制结果完全相同。当图像分辨率为 1 024×1 024 时,对于比较复杂的数据,例如头部(血管),加速性能有所下降,然而对于空体素比例较高、非空体素比较集中的数据如体部(骨骼),加速性能有进一步的提高。这也正反映了本算法的本质特点:快速跳跃空体素。

5 结 论

长期以来医学图像的三维重建一直是图形工作站的天下,基于 PC 平台的三维重建软件由于绘制时间长或者图像质量差而无法在实际诊断中应用。由于图形工作站价格昂贵,又需要特殊的软件平台,且大部分软件平台都与专用数字医疗设备相配套,因此医学图像三维重建在辅助诊断中的应用比例极低。随着医院 PACS 系统的推广,医生已经可以很方便地在自己的 PC 终端上获取图像,因此他们迫切地需要可以运行于价格低廉的 PC 机平台并且可以获得高质量绘制结果的三维重建软件,以便于在离线状态下进行辅助诊断和医学技术的培训。

该加速算法可以在保证图像质量的前提下较大幅度地提高体绘制的速度,基本上可以满足用户实时交互的需求。当然目前的算法还存在一定的缺陷,本文加速算法的核心思想是快速跳跃空体素,当空

体素的比例较小时,算法的加速性能将严重下降,同时不同的观察角度的绘制时间差异较大,给用户的交互带来不便,进一步的研究将致力于这一问题的解决。同时笔者也认识到,纯粹的软件加速其加速性能还是有限的,虽然处理器的速度在飞速地提高,但是需要处理的数据量也在急剧增强,因此结合图形硬件的图形加速能力,实现软硬件的综合加速将是最终的解决之道。

参 考 文 献

- 1 Laszlo S K, Vlastimil H, Benedek B, *et al.* On the efficiency of ray-shooting acceleration schemes[A]. In: Proceedings of ACM SCCG'02[C], Budmerice, Slovakia, 2002; 89~98.
- 2 Yagel R. Towards real time volume rendering [A]. In: Proceedings of GRAPHICON' 96 [C], Saint-Petersburg, Russia, 1996; 230~241.
- 3 Levoy M. Volume rendering by adaptive refinement[J]. The Visual Computer, 1990, 6(1): 2~7.
- 4 Pyokkimies E P, Salli E, Katila T. Fast image order volume rendering algorithm for multimodal image visualization[A]. In: Proceedings of 12th International Conference on Biomagnetism [C], Espoo, Finland, 2000; 1043~1045.
- 5 Walsum T V, Hin A J S, Versloot J, *et al.* Efficient hybrid rendering of volume data and polygons [A]. In: Advances in Scientific Visualization [M], Berlin: Springer-Verlag Press, 1992; 83~96.
- 6 Yagel R, Kaufman A. Template-based volume viewing [J]. Computer Graphics Forum, 1992, 11(3): 153~157.
- 7 Cohen D, Sheffer Z. Proximity clouds, an acceleration technique for 3D grid traversal[J]. The Visual Computer, 1994, 11(1): 27~38.
- 8 Semwal S K, Kvarnstrom H. Directed safe zones and the dual extent algorithms for efficient grid traversal during ray tracing [A]. In: Proceedings of Graphics Interface'97[C], Ke-lowna, Canada, 1997; 68~75.
- 9 Yagel R, Shi Z. Accelerating volume animation by space-leaping [A]. In: Proceedings of IEEE Visualization'93[C], California,

USA, 1993: 62~69.

10 Brady M L. Interactive volume navigation [J]. IEEE Transactions on Visualization and Computer Graphics, 1998, 4(3): 243~256.

11 Chen B Q, Kaufman A, Tang Q Y. Image-based rendering of surfaces from volume data[A]. In: Proceedings of International Workshop on Volume Graphics[C], New York, USA, 2001: 279~295.



徐心和 1940年生,东北大学人工智能与机器人研究所,教授,博士生导师。研究领域为智能机器人、人工智能、模式识别、智能控制。



胡英 1977年生,2001年于东北大学控制理论与控制工程专业获硕士学位,现为东北大学人工智能与机器人研究所博士生。研究领域为科学计算可视化、虚拟现实和机器人视觉。