

虚拟现实的图形生成技术*

刘学慧 吴恩华

(中国科学院软件研究所计算机科学实验室, 北京 100080)

摘要 虚拟现实是一种高度逼真的模拟人在自然环境中视、听、动等行为的人机界面。图形生成是虚拟现实技术的重要瓶颈。本文对面向虚拟现实的实时图形生成技术及其发展情况作了详细的介绍和综述。其主要内容是图形生成的硬件体系结构以及在虚拟现实的真实感图形生成中用于加速的各种有效技术。

关键词 虚拟现实, 图形生成, 图形加速

随着处理器技术的大幅度提高及图形绘制技术、数字信号处理技术、传感技术、图形技术的发展, 80年代末、90年代初国际国内形成了对虚拟现实的研究热潮¹⁾。所谓虚拟现实技术即利用计算机产生逼真的三维视觉、听觉、触觉等感觉, 使得用户可以通过使用专用设备自然地对虚拟环境中的实体进行交互考察与控制¹⁾。

虚拟现实技术作为一种高度逼真地模拟人在自然环境中视、听、动等行为的人机界面技术, 具有两个基本特征, 即交互和身临其境。其中, 身临其境特征要求计算机所创建的三维虚拟环境看起来、听起来和感觉起来是真的。无疑, 使用户对虚拟环境产生犹如身临其境的可行性技术的应用是虚拟现实走向成功的必要条件。这其中包括先进的交互、视频等技术的发展和运用。而对于虚拟现实的虚拟环境而言, 实时动态的图形视觉效果是产生现实感觉的首要条件, 舍此便无“现实”可言。因而图形快速生成技术便成为虚拟现实的关键技术²⁾。

1 图形生成是虚拟现实的重要瓶颈

虚拟现实最重要的特性是人可以在随意变化的交互控制下感受到场景的动态特性。换句话说, 虚拟现实系统要求随着人的活动(位置、方向的变化)即

时生成相应的图形画面。有两种重要指标衡量用户所沉浸于虚拟环境的效果和程度。其一是动态特性, 其二是交互延迟特性。自然的动态特性要求每秒生成和显示 30 帧图形画面, 至少不能少于 10 帧²⁾, 否则将会产生严重的不连续和跳动感。交互延迟是影响用户感觉的另一个重要指标。对于人产生的交互动作, 如飞行模拟时飞行位置、方向的控制, 系统的图形生成必须能立即作出反应并产生相应的环境和场景。其间的时间延迟不应大于 0.1 秒, 最多不能大于 1/4 秒³⁾。否则在长期的工作中, 人会产生疲劳、烦躁甚至恶心的感觉, 严重地影响“现实”的效果。以上两种指标均依赖于系统生成图形的速度。对于动态图形效果而言, 每帧图形的生成时间局限于 30~50 毫秒; 而对于交互延迟, 除包含对于交互输入及其处理时间外, 图形生成速度亦是重要因素。显而易见, 图形生成速度是虚拟现实的重要瓶颈³⁾。图形生成的速度主要取决于图形处理的软硬件体系结构, 特别是硬件加速器的图形处理能力, 以及图形生成所采用的各种加速技术。除此之外, 还依赖于应用的因素, 虚拟场景的复杂程度和图形生成所需的真实感程度。本文将在第二节介绍虚拟现实系统所具有的几种典型系统结构, 而后在第三节介绍虚拟现实图形生成中用于加速的各种有效技术。

* 本文工作得到国家自然科学基金的资助 收稿日期: 1996-07。

1) 石教英. 虚拟环境(VE)及其应用. 见: 虚拟现实专家研讨会(自然科学基金委员会主)文集, 杭州, 1994年11月。

2) 吴恩华. VR中三维图象绘制及有关加速技术(同上)。

3) 吴恩华. 论图形生成和虚拟现实. 见: 中国科学院软件研究所学术报告集. 第2卷, 1995年4月。

2 虚拟现实图形绘制的系统结构

就图形学发展而言,起关键作用的无疑是图形硬件加速器的发展。计算机图形硬件和软件系统在 60、70 年代以计算机的一种外部设备和可选支撑系统的面目出现。进入 80 年代后,图形硬件加速器首先作为当时出现的计算机新机种——图形工作站硬件系统的必备功能部件而得到推广。图形处理与显示性能成为衡量工作站性能的重要指标。

随着计算机图形学在各领域中应用的不断扩展,人们对图形质量、图形生成速度要求的不断提高,单一流水线结构的图形工作站不再能满足人们的需求。据 Molnar^[4]估计,以每秒 30 帧的画面生成速度绘制一个具有 10 万个多边形的场景,图形工作站的硬件速度必须达到每秒 300 兆浮点计算、750 兆的整数计算以及 450 兆次对帧存储器访问的能力。要满足人们不断增长的要求,图形生成的并行处理势在必行。

图形生成硬件流水线由几何处理(包括视角变换、投影及裁剪等)和绘制两部分组成。图形生成的并行处理主要利用了这两个阶段处理的图形单元的无关性,通过多重相同结构的硬件处理器来提高总体运行速度。在几何处理阶段,根据各种分配策略,采用 SIMD 或 MIMD 的体系结构,将各图形处理单元均衡地分布在各处理部件上。在绘制阶段,由于帧缓冲存储器存取带宽是制约绘制速度的主要瓶颈,现有的并行结构大多利用商业化高速缓冲存储器交错存取存储器,为各组存储单元分配自己独有的绘制部件以达到高的并行处理性能。以上在图形生成两阶段实行的并行处理在各并行图形工作站上往往是不相关的。北卡大学的设计者们认为要达到高性能的图形处理,必须将两者有机地结合起来。基于这种思想,北卡大学推出了两种大规模并行图形工作站 PIXEL-PLANE 系列机和 PIXELFLOW。1992 年推出的 PIXELFLOW^[4]利用图象合成技术,将由不同处理机绘制的部分场景的全屏幕图象合成为整个场景的图形画面。整个合成过程由硬件实现。PIXEL-PLANE 系列机则是 80 年代初开始推出的大规模图形生成并行处理工作站。其主要思想是利用物体的边界方程^[5]将屏幕空间划分成可并行处理的象素块,并行计算各象素点的值。该大学于 1990 年推出的 PIXEL-PLANE5 具有每秒绘制 1 兆具有 phong 光照模型的多边形的处理能力^[6]。考虑到物

体划分不能太小,该系统将 1024×1280 个象素点组成的屏幕空间划分成由 128×128 个象素点组成的多个虚拟子空间,落在各子空间范围的物体由多个并行绘制处理器并行绘制,最后拼合为一个全屏幕的图形画面。目前该系统已被应用于一些 VR 系统。其中包括北卡开发的名为 Nanomanipulator^[7](纳米操作装置)的 VR 系统。该系统将高精度(纳米级)检测设备所采集的被观察表面的数据转化为用户定义的数量级数据,同时实时生成所观测表面的画面。观察者可根据观察结果对被观测表面进行实时修改,并将检测设备实时采集的经过表面修改后的数据反馈给图形绘制工作站。该系统利用 PIXEL-PLANE5 可对多个图形处理器直接编程的能力,将反馈结果对图形画面的修改局限于局部范围,从而实时生成观察者对观测表面进行了修改后的画面。该系统具有很高的实用价值,如使用在纳米级的电子线路设计中等。

虽然高度并行的图形处理硬件可具有相当强大的图形处理能力,但如何使各种应用问题充分地利用并行处理机的并行性常常是一个很大的难题。这意味着如果不能充分地利用其并行处理能力将是对资源的浪费。在实际应用中,大多数的 VR 系统仍然采用现有的图形工作站作为其图形绘制的平台。

在当前三维图形工作站的群雄竞争中,SGI 工作站作为最杰出的代表具有最优越的三维图形功能。事实上,SGI 工作站的发展历程正是代表着研究部门和工业界 10 多年来在图形硬件方面为计算机图形处理的发展所作的艰苦卓绝的努力。在 80 年代初,正是 SGI 公司的创始人 James Clark 教授将三维图形生成的最基本功能——几何变换的矩阵运算完全使用大规模集成电路加以实现,从而为三维图形流水线的主要功能,即坐标变换、三维投影和图形裁剪等的实现奠定了硬件实现的基础。在此基础上,一代又一代升级的图形硬件加速器为 SGI 的系列工作站提供了越来越快速的图形处理能力。甚至某些重要而高级的图形特性,如纹理映射、景深等功能也可由硬件实现^[8],使得诸多领域如地理地貌、建筑漫游、科学计算可视化等 VR 系统的实时环境具有更强的真实感。当今 SGI 最先进的图形加速器已具备每秒绘制超过 1 兆具有浓淡光照的多边形的能力。为此,最典型的 VR 系统常常是使用高档的 SGI 工作站作为其实时图形绘制的处理机,而使用单独的微机作为输入/输出的交互处理。二者用 Ethernet 等局部网联接。当需要为双眼产生不同的立体图象

时,往往用两台相同的工作站或多处理机的两台处理机分别作左、右眼的图形生成。使用头盔式显示器作为其视觉环境的虚拟现实系统通常采用这样的硬件系统配置。

除采用头盔式显示器作为视觉环境外,虚拟现实系统另一种典型视觉环境是创造一种用户身在其中的影视环境。Illinois 大学设计实现的 CAVE 系统^[9,10]是这种 VR 系统的典型代表。系统开发者从环幕电影得到启发,利用今天高性能的图形工作站的平面投影绘制能力和高精度的投影设备,设计出这样一种基于投影的无障碍式、融真实世界与虚拟世界为一体的 VR 系统。CAVE 的可视化环境是由 4 个包围着观察者的投影面(前、后、左、右)组成的四方形空间。系统配备 5 台 SGI 工作站,其中 4 台完成 4 个投影面的图形生成;另外一台负责信息传输和各处理机的协调工作。观察者只需配备轻便的立体眼镜和头部跟踪器就能感受到投影图象充满着空间。随着观察者在此空间的漫游,观察者可以观察到周围“环境”的动态变化,并能观察到自己的身体,给人以较“真实”的沉浸感。同时,系统根据观察者离屏幕的远近,由远端超级计算机实时生成逐步求精的科学可视数据,并由 SGI 完成绘制。CAVE 已成功应用于各科学可视化系统中。其中由 Illinois 大学开发的用于观察美国国家航天航空局卫星观测数据 FIFE 的基于 CAVE 科学可视化环境的 VR 系统 SANBOX^[11]成功地为人们展示了曼哈顿周围 20×20 平方公里 120GiGa 的气象卫星。

允许多个用户同时进入的虚拟网络系统是虚拟现实的另一种典型体系结构。这种类似于客户/服务器的虚拟网络分布式系统,将各种任务的模拟分散在由 LAN 或 EtherNet 连接的多个工作站上。该种系统结构不仅能充分利用已有的设备,而且便于远端计算机的加入,以允许多用户同时进入虚拟环境。例如大规模作战人员参加的虚拟战场演示环境就是这种分布式体系结构的典型应用环境。1989 年,美国国家先进技术研究总署基于陆战演示分布式系统 SIMNET^[12]开发了虚拟飞行作战演视系统 DIS^[13]。该系统由位于美国本土及加拿大的多个虚拟座舱通过网络连接而成。虚拟座舱由一台 SGI 公司的多处理器图形工作站 4D/440VGTX 通过光缆与头盔显示器、头部跟踪器及控制杆连接而成。由于 UNIX 系统的分时特性,使得系统对飞行的访问时间间隔较长,系统将动力模型参数更新进程的访问时间间隔划分为 20 个更小的时间间隔。同时,系统

分配不同的处理器分别管理飞行动力模拟与绘制和网络界面进程,以加快系统对飞行动力模型的访问。为减少网络传输量,降低由于网络传输带来的延迟,系统根据预测推算算法(Dead Rechoning)发送消息,并根据最近接收的远端虚拟器的消息来推断其位置和状态。

虚拟现实系统的软硬件结构涉及到如何有机地协调多输入/输出、动态模拟及图形综合等多任务。系统在处理多种输入输出以及大量模拟计算的同时,必须使用户在视点改变后即时感受到场景的变化,这是虚拟现实系统是否成功的决定性因素之一。为保持较高的帧频率,虚拟现实系统通常将各进程放置在异步运行的处理机上。系统必须协调处理好各进程之间的关系,否则系统对用户交互的反应将会受到严重的影响。便如 VPL 公司的 Electric/ISAAS^[14]因为输入设备交由应用计算进程管理,尽管绘制进程和应用进程分置在不同的处理机上,绘制进程却必须等待应用计算进程的信息才能反应用户的交互动作,使得绘制进程受限于应用计算进程的更新率。为改进对用户交互的实时反应,Alberta 大学开发的虚拟环境建模系统 MR^[15]将计算进程和应用进程放置于不同的处理机上,并将绘制等输入/输出进程交由应用进程管理。但该系统不提供应用进程和计算进程的协调工作,用户必须花很大精力对二者进行协调。与此相对照的是, Virginia 大学开发的虚拟环境建模系统 DIVER^[16]将输入设备直接与绘制数据库相连,使得系统在设计应用计算进程进行复杂计算的同时,用户仍能随视点的改变看到连续动态的画面,以伴随虚拟场景的缓慢变化。同时系统提供放置于不同处理机上的应用计算进程和绘制进程的“透明”性协调工作,极大地方便了虚拟现实系统的开发。

3 图形生成的加速方法

虽然得益于高速发展的 CPU 和专用图形处理器,使得现今的图形工作站性能得到很大的提高,但距离 VR 的需求仍相当遥远。换句话说,当前图形生成的速度相对于通常 VR 环境的规模来说仍存在相当大的差距。举例来说,UC. Berkely 大学的计算机科学大楼的计算机模型具有 1.4 兆个多边形的数据;帝国大厦的计算机模型则由 167 兆个多边形组成;而波音公司最近推出的完全由计算机设计的波音 777 是一个有着 500 兆个多边形的计算机建模模

型。这些应用模型的复杂程度往往超过当前图形工作站的实时处理能力。考虑到 VR 对场景复杂度几乎无限制的要求,在 VR 高质量图形的实时生成要求下,如何从软件着手,减少图形画面的复杂度,已成为 VR 图形生成的主要目标。

下面我们就一些实时图形生成加速方法作一介绍和分析。

3.1 可见性的判定

由于视线的方向性、视角的局限性以及物体相互遮挡,人眼所看到的往往只是场景中的一部分,而图形的生成是物体空间到图象空间的转换。为充分利用绘制硬件的有限资源,就必须充分利用物体空间的相关性。图象空间的相关性和时间空间的相关性可以加快可见性的判定,减小绘制深度。

对一个非常复杂的场景,物体的细节描述包含大量的多边形,这就需要一个有效的数据结构来组成场景以加快可见性的判定。70 年代末发展起来的物体分层次表示(Hierarchical Representation)大大地加快了可见性的判定。层次表示的主要方法是包围盒技术和八叉树技术。这两种方法的主要特点是将场景组织成一棵树,充分利用空间连贯性以加速场景的遍历,从而大大地减少了画面绘制过程的空间复杂度。Fuchs^[17]以物体的半空间表示将其空间一分为二,从而将场景组织为二叉树(BSP)的结构。此方法可利用已建立起来的二叉树很快地决定相对视点面的前后排列。但该方法将产生大量新划分面片。鉴于以上方法的特点,Teller^[18]基于建筑 CAD 的特点,探索了一种基于“房间”结构的可见性预处理方法。此算法以建筑模型中主要的遮挡面作为空间划分面,从而将建筑模型划分为以房间为基本单位(cell)的数据模型。在空间划分的基础上,根据基本单位之间共有的可见口(portal),如门、窗等,建立起 cell 与 cell 之间的邻接图。考虑到实时漫游时观察点的任意性,算法定义 cell 的可见性为观察者所能看到的区域,即一个观察者在此 cell 中的任一位置、任一方向上所能看到的区域。由于 cell 与 cell 之间可见就意味着有一条光线能无遮挡地从此 cell 达到彼 cell。算法根据 cell 与 cell 之间的邻接图,通过深度遍历建立起 cell 与 cell 的可见口序列(portal sequence),从而得到 cell 对 cell 的可见性超集、cell 对 object 的可见性超集。在实时漫游时,算法利用预计算的结果实施内存与外存的调度管理,并从所得的超集计算视点对 cell、视点对 object 的

可见性集合,从而有效地加速了可见性的计算。但本算法的预计算比较复杂,且目前局限于轴向的建筑模型。

不论哪种空间划分技术,这些方法都仅仅是利用物体空间的相关性来加快可见性的判定,传统的 Z-buffer 算法利用同一象素的深度信息,决定可见面的判定,有效地利用了图象空间的相关性来加快可见性的判定,但 Z-buffer 算法却没有做到这点。在利用物体空间、图象空间和时间空间的相关性方面,Greene^[19]提出了非常有创意性的思想。该方法利用绘制画面的 Z 缓冲存储器,通过物体空间和图象空间两个结构的建立,提出了一个利用物体空间、图象空间和时间空间的相关性的算法。我们知道,八叉树中一个单元如其六个面不可见,则此单元所包物体不可见。那么如何判定此单元不可见?Greene 利用已绘制场景的 Z 缓冲存储器结果作为 Z 缓冲存储器层次结构的第一级,将上一级数据压缩一倍(即取每相邻四个数据的最大值),就得到第二级数据。依此压缩下去,就得到图象空间的层次结构。可以看出,层次结构中的任一数值都对应于屏幕空间上不同区域的最大 Z 值。以八叉树中一单元在屏幕空间所在区域的最大 Z 与此单元的最小 Z 值比较,如果此屏幕空间的最大 Z 值比此单元的最小 Z 值小,那么此单元不可见。否则将此单元分为四个子空间,再将这四个子空间的最小 Z 值与所对应的屏幕空间区域的最大 Z 值分别进行比较,直到子空间的最小 Z 值大于相应屏幕空间区域的最大 Z 值或者相应屏幕空间区域已对应于屏幕一象素。虽然此算法的实施依赖于可否对 Z 缓冲存储器进行访问,但其方法在利用物体空间、图象空间和时间空间一致性方面作了很有效的尝试,算法的实施非常便利于并行计算。

3.2 细节层次模型

为物体提供不同的细节层次描述是控制场景复杂度的一个非常有效的方法。在生活中我们知道,当物体离得越来越远时,人们不再能辨清该物体上的许多细节结构,如光滑的曲面。因此,绘制一个远处的物体时,用该物体细节描述非常复杂的模型是完全没有必要的,也势必白白浪费图形处理资源和处理时间。早在 1987 年,Clark^[20]就指出,当物体仅覆盖屏幕较小区域时,可以用该物体描述较粗的模型进行绘制,以便达到快速、有效的复杂场景绘制。Crow^[21]用实例给出了同一物体采用不同的细节层

次模型描述所具有的优点,并指出应自动生成同一物体的多层次细节模型。

基于各种曲线曲面拟合的物体模型可以自动生成该物体的不同层次的细节描述。而对于来自各种应用的基于多边形网格模型的多细节层次模型的自动生成也已有广泛的研究,主要有以下几类方法,这里分别给以简单介绍。

自适应递归划分是曲面曲线生成的常用方法。Schmitt^[22]将其与曲面拟合技术相结合,用不断逼近的曲面片逼近所要简化的物体模型,为规则四边形网格表示物体建立起基于 Bezier 曲面表示的简化模型。该算法首先为该规则四边形网格表示的物体建立了一个分片式双三次 Bezier 近似曲面。对每个局部曲面片,如果该曲面片在用户定义的误差内已近似地逼近该面片所表示物体的部分,则算法停止;否则,将该面片划分为四个子曲面片,并依一阶几何连续要求建立起这四个子曲面片控制点的方程,根据物体局部几何信息,决定控制点的选取。该算法不仅自动生成用规则四边形表示物体的简化模型,而且有较为精确的误差控制。基于此方法,Dehaemer^[23]提出了基于规则四边形网格表示物体的简化多边形模型的自动生成方法。

Greg^[24]给出了基于网格重新划分的多边形模型简化方法。该方法首先将形成新的细节层次模型所需的新顶点在原模型上重新布点。为给曲率变化较大的区域分布更多的新顶点,以使重新划分的三角形网格模型较精确地反映出原模型,该算法根据各顶点的曲率近似计算,用三角形顶点曲率与三角形面积的乘积因子决定新顶点的放置,并根据顶点间的排斥半径将新顶点定位在原多边形内。然后,根据新旧顶点形成中间网格模型。通过必要的拓扑结构一致性检查移去老顶点,并进行局部的三角形化操作,形成新的符合原物体拓扑关系的网格模型表示。

基于顶点删除的多边形消减方法是自动生成物体简化模型的另一种有效方法。其中 Schroeder^[25]首先利用各顶点的局部几何和拓扑信息将各顶点分类,然后根据不同类顶点的评判标准决定该顶点是否可以删除。如果可删,则采用递归循环分割法对删除顶点后所遗留的空洞进行三角化剖分。否则该顶点作为简化模型中的顶点保留。

Kalvin^[26]给出了面片合并方法自动生成多边形表示物体的简化模型。该方法首先以原模型中的任一多边形为“种子”面片,以一组合并条件不断合并

种子面片周围的面片,直到周围面片不再满足合并条件。同时,算法根据用户定义的误差给出种子面片的近似面片集合,并根据不断合并的种子面片修改近似面片集合,直到集合为空。

以上方法都要求生成的简化模型保持拓扑关系和几何关系的正确性。但 Rossignac^[27]认为,简化后的模型只要其图形生成效果与原模型一致,则该简化模型成立。为加快实时图形生成,该文提出了一种多精度近似法(Multi-resolution Approximation)自动生成物体的简化模型。该方法首先赋予各顶点以重量值,给物体特征变化较大处的点以较大的重量值。然后根据模型的复杂程度、相对大小等将此物体所占空间划分为立方体的 cell。同一 cell 中的顶点,以各顶点的重量值加权计算这一 cell 中所有顶点的代表点,然后依据原模型中各多边形顶点的代表点是否为同一代表点合并各多边形。

有了物体的细节层次模型之后,如何选择物体当前的层次描述呢?根据人的视觉特征,主要依据以下标准:①根据物体与视点的距离及物体在投影平面所占空间的大小来选择,这是最常用的方法;②根据物体与视线方向的夹角来选择,因为视网膜中心的成象看得更清楚,而周围视觉则更易觉察物体的动感;③根据人与物体是否存在相对运动来决定物体的精细程度的描述。由于运动的突发性,如何作到不同精细模型的光滑过渡仍有待于进一步解决。

光滑过渡不仅是依据运动判断物体细节层次模型的问题,也是细节层次模型关注的课题之一。为解决这一问题,目前所采用的方法是 Eading 和 Morphing。

在不同层次判定距离时,Fading 将物体看作是这两个不同层次模型的混合(Blending),即利用图形硬件的透明绘制效果,赋予不同层次模型相应的透明指数,同时绘制这两个不同层次模型。虽然这种方法很大程度上解决了突变问题,但由于必须在绘制一帧画面的时间内同时绘制同一问题的两个不同细节层次的模型,其时间耗费较大,并且图形硬件实现的透明并不能充分表示混合效果。Morphing 利用动画变形方法,建立不同层次模型间物体的对应关系,中间插值得到中间物体的描述,较好地解决了突变问题,其效率较高。

以上讨论的都是对网格化的小范围曲面物体提出的细节层次模型的建立及选择方法。但对于像地表模型这样的大面积物体,因为视点看到的仅是物体的一部分,以上的方法便不再适用。虽然可将地表

模型划分成小面积的区域,从而分别为各区域建立细节层次模型并选择相应的层次描述,但由于存在相邻区域不同层次间的“裂缝”,使得这一方法存在难以克服的困难。现行较为有效的方法是利用改进的分形空间位移插值法^[28],实现局部区域进一步的细化,以此解决地表模型的细节层次模型的建立和实时显示问题。

3.3 预测计算

预测计算是一种减少系统延迟的方法。该方法根据各种运动的速率和加速度,如人体头部转动速度,用预测、外推法在下一帧画面绘制之前估算出头部跟踪系统及其它输入设备的输入,从而减少由输入设备所带来的延迟。对人体测试表明,人体头部转动的最大速率为 100 度/秒^[3],那么可以预测,到下一帧画面,人的视线方向变化大小不超过 3~5 度。虽然预测方法根据实时所测数据可以有效地解决延迟问题,但系统对用户动作的正确反应依赖于正确的预测,据此在实际应用中,人们往往利用外插法所得的各种模拟数据的变化,如观察者的视点位置、凝视方向、各种运动问题的位置及方向来减少延迟。如系统根据预测猜定用户视点位置和视线方向,但在依据视见体判定可见面时,系统依据人体头部转动速率的测试结果将通常的视见体角度左右加大 5 度。在依此加大视角的视见体的可见性判定后,系统才根据这时外设输入的较精确视点位置和方向绘制图形。这一方法不仅克服了输入设备所带来的延迟问题,而且解决了由预测所引起的错误。

在分布式网络虚拟现实系统中,系统根据预测估算原理来减轻网络传输负载并减少由网络传输带来的延迟。在前面介绍的 DIS 系统中,本地模拟器只有在不能确定自身方向、速度时,才向网络发送有关自身状态的消息。同时,本地模拟器根据最近接收的远端模拟器的消息来推算远端模拟器的位置等状态。

3.4 脱机计算

脱机计算是减轻计算机实时计算负载的常用方法之一。由于 VR 系统是一个多任务的模拟系统,所以有必要尽可能地将一些可预先计算好的结果预先计算并贮存在相应的结构中,其中包括全局光照模型、动态模型的计算等。

就图形生成质量而言,VR 所期待的目标自然是高质量的具有真实感的逼真图形,并具有各种自

然场景的复杂特性的画面。但如前所述,特性处理的硬件能力尚远远满足不了 VR 的需求,因而在现今的条件下,实时的真实感特性生成几乎是完全不可能的。对于大多数的 VR 应用来说,仍是采用简单的局部光照模型,且在硬件部件的支持下实现质量并不高的特性显示。但从将来的发展来看,使用全局光照模型和显示具有复杂场景的真实感图形必然是 VR 所应追求的目标。就全局光照模型来说,光线跟踪技术由于其计算量大,且因其生成结果固有地与视点相关而很难进行预计算,因而这一技术几乎很难应用于 VR。与此相对,辐射度技术却因其计算结果与视点无关,大量计算可作预计算,因而最有希望成为用于真实感图形生成的工具。

自 1984 年 C. M. Coral 等从热辐射工程中引进了辐射度方法之后,辐射度技术已得到了很大的发展。特别是逐步求精算法的提出,大大加快了辐射度技术发展的速度,使得辐射度技术成为真正实用的全局光照模型技术。进入 90 年代后,辐射度技术进一步走向实用。多层次划分技术^[29]利用 N-body 问题的方法将传统辐射度方法的时空复杂度从 $o(n^2)$ 降为 $o(n)$,并根据用户定义的误差,自动建立起面片的划分,进一步提高了辐射度技术的实用性。1994 年,Smits^[30]在一文中将视点参数引进辐射度计算中,建立起与视点范围有关的 importance 函数,将视点与辐射度技术结合起来,使辐射度技术向实时计算迈进了一步。

物体的动态模拟模型的计算是 VR 的一大任务。在这些动态模拟中包括一些较有规律的动态特征的模拟,如具有动画特征的随时间变化而变化的动态模型和一些基于特定事件的动态事件的模拟。在它们的物理模型中的各种参数并不因各种偶然性而发生变化,如物体的独自运动。为减少实时动态模拟计算的负载,可以将这些动态描述作脱机计算,并作为实时计算的输入。在德国 Fraunhoffer 计算机图形研究所为工业应用设计的 VR 应用系统 Virtual Design^[31]中,利用辐射度计算和动态模拟的脱机计算已有了较为成功的尝试。

Virtual Design 是一个面向建筑设计的开放式系统。该系统实现多种 CAD、室内设计等的图形文件到自身数据库的转变。系统为满足不同需求,运用了不同精度要求的辐射度算法预计算虚拟场景的全局光照模型。系统设计者们认为两层数据结构已足够,多层则不利于绘制速度。为减少动态模拟的实时计算,系统将一些动态模拟作脱机计算,并以矩阵形

式作为系统的输入。该系统提供多种输出,其中包括三维真实音响的模拟。用户可感受到在虚拟场景中漫游时不同地点的音响特质。

以上介绍的是各种常用的图形生成加速技术。当然,图形生成的速度还有赖于其它一些因素,如数据库的结构等。数据库不仅是虚拟现实对虚拟环境的描述所在,而且它的数据结构还关系到VR系统的真实效果。就图形描述而言,为实时的数据库遍历提供良好的结构,减少实时可见性计算量等亦是VR系统成功的关键因素。为充分利用图形硬件资源,避开图形处理的瓶颈,数据库将尽量将三角面片组织为共顶点的三角面片串(mesh)。同时,为减少图形属性的改变,如光照特性等的变化,有些VR系统将面片按其属性进行组织,以减少特性属性变化所带来的时间耗费。为减少图形硬件的矩阵计算,数据库应尽量预先计算好各静态物体的矩阵变化。

参考文献

- 1 Carolina Cruz-Neira. Virtual Reality Overview. SIGGRAPH'93, Course Notes 23.
- 2 Steve Bryson. Virtual Reality in Scientific Visualization. IEEE Computer&Graphics, 1993, 17(6).
- 3 Michael Jones. Lesson Learned from Visual Simulation. SIGGRAPH'95, Course Notes 6.
- 4 Stevem Molnar. Pixelflow: High-Speed Rendering Using Image Composition. SIGGRAPH'92, 26(2): 237~240.
- 5 Goldfeather, Jack, Henry Fuchs. Quadratic Surface Rendering on a Logic-Enhanced Frame-Buffer Memory System. IEEE CG &A, 1986, 6(1): 48~59.
- 6 Henry Fuchs. Pixel-Plane5: A Heterogeneous Multiprocessor Graphics System Using Processor-Enhanced Memory. SIGGRAPH'89, 23(3): 79~81.
- 7 Russell M, Talor II. The Nanomanipulator: A Virtual-Reality Interface for a Scanning Tunneling Microscope. Proceeding of SIGGRAPH'93: 127~133.
- 8 Paul Haeberli. The Accumulation Buffer, Hardware Support for High-Quality Rendering. Computer Graphics, 1990, 24(4): 309~318.
- 9 Carolina Cruz-Neira. Scientist in Wonderland: A Report on Visualization Application in CAVE Virtual Reality Environment. IEEE Symposium on: Research Frontiers in Virtual Reality, 1993: 59~66.
- 10 Carolina Cruz-Neira Surround-Screen Projection-Based Virtual Reality: the Design and Implementation of the CAVE. Computer Graphics Proceeding Annual Conference Series, 1993: 135~142.
- 11 Andrew Johnson, Farshad Forouhi. SANBOX: Interface to Scientific Data Based on Experimentation. Fifth Eurographics Workshop on Visualization in Scientific Computing, 1994.
- 12 Thorpe J A. The New Technology of Large Scale Simulator Networking: Implication for Mastering the Art of Warfighting. Pro. Ninth Interservice/Industry Training System conf., Defense Technical Information Center, 1987, 11: 492~501.
- 13 Deen McCarty W. A Virtual Cockpit for A Distributed Interactive Simulation. IEEEECG&A, 1994: 49~54.
- 14 Blanchard, Chuck. Reality Built for Two: A Virtual Reality Tool. ACM SIGGRAPH 1990 Symposium on Interactive 3D Graphics, 1990: 35~36.
- 15 Green, Mark. Minimal Reality(MR): A Toolkit for Virtual Applications, MR v1.0, Programmer's Manual, University of Alberta, 1991.
- 16 Rich Grossweiler. DIVER: A Distributed Virtual Environment Research Platform. IEEE Symposium on: Research Frontiers in Virtual Reality, 1993: 10~15.
- 17 Fches H. Z. Kedem, and B. Naylor. On Visible Surface Generation by APriori Tree Structure. Computer Graphics (Proc. SIGGRAPH'80), 1980, 14(3): 124~133.
- 18 Seth J. Teller. Visibility Preprocessing for Interactive Walk-throughs. Computer Graphics, 1991, 25(4).
- 19 Nech Greene. Hierarchical Z-Buffer Visibility. Computer Graphics Proceedings, 1993: 231~238.
- 20 Clark, James H. Hierarchical Geometric Models for Visible Surface Algorithm. Communications of the ACM, 1976, 19(10): 547~554.
- 21 Crow, Franklin C. A More Flexible Image Generation Environment. Computer Graphics, 1982, 16(9): 9~18.
- 22 Francis J M Schmitt. An Adaptive Subdivision Method for Surface-fitting from sampled Data. SIGGRAPH'86, 1986, 20(4): 179~188.
- 23 Dehaemer W J. Simplification of Object Rendered by Polygonal Approximations. Computer Graphics, 1991, 15(2): 175~184.
- 24 Turk Greg. Re-tiling Polygonal Surface. Computer Graphics Proceedings, 1992: 55~64.
- 25 Schroeder W J. Decimation of Triangle Meshes. Computer Graphics, 1992, 26(2): 65~70.
- 26 Alan D. Kalvin. Supersurface: polyonal Mesh Simplification with Bounded Error. IEEE Computer Graphics and Applications, 1996: 64~77.
- 27 Jarek Rossignac. Muti-Resolution 3D Approximation for Rendering Complex Scenes: 455~465, in Geometric Modeling in Computer Graphics, Springer, Verlag, Eds. B. Falcidieno and T. L. Kunii, Genova, Italy, 1993-06: 28~1993-07: 2.
- 28 R. L. Fergusen. Continuous Terrain Level of Detail for Visual Simulation. Proceeding of the 1990 IMAG V Conference, 1990, 6: 145~151.
- 29 Pat Hanranhan. A Rapid Hierarchical Radiosity Algorithm. Computer Graphics, 1991, 25(4): 197~206.
- 30 Brain E. Smits. An Importance-Driven Radiosity Algorithm. Computer Graphics, 1991, 25(4): 197~282.
- 31 Peter Astheimer. Virtual Design: A Generic V. R. System for Industrial Application. Computer&Graphics, 1993, 17(6): 671~678.



刘学慧,1994 年于湖南湘潭大学获硕士学位。现为中国科学院软件研究所博士研究生,主要从事专业为计算机图形学。

Techniques for Graphic Generation in Virtual Reality

Liu Xuehui, Wu Enhua

(Laboratory of computer science, Institute of Software, CAS, Beijing 100080)

Abstract Virtual Reality is an advanced man--machine interface which allows the behavior of the human beings, such as viewing, listening and touching, to be conducted in a natural but virtual environment. Graphic rendering is the bottleneck in VR. This paper gives a detailed description of realtime graphic generators associated with Virtual Reality. The primary topics involved are in two aspects: the structure of hardware systems for graphic generation and various methods for accelerating graphic rendering.

Keywords Virtual reality, Graphic generation, Accelerating graphic rendering