

2000, 27(11)

分布式虚拟环境

计算机科学 2000 Vol. 27 No. 11  
虚拟现实 人机交互

## 分布式虚拟环境综述\*

Distributed Virtual Environment Overview

赵沁平 何红梅

TP391.9

(北京航空航天大学计算机科学与工程系 北京 100083)

**Abstract** Distributed virtual environment consists of human-machine interaction, three-dimensional virtual environment, entity modeling and network communication. In this paper, key technologies in current systems are presented.

**Keywords** Virtual reality, Virtual environment, Distributed virtual environment

## 1 引言

虚拟环境是指用户借助各种直观的输出输入设备,沉浸于计算机生成的真实的或虚幻的三维世界中,漫游其中或与其中的对象交互,从而获得如同在真实物理世界中的体验<sup>[1]</sup>。虚拟环境也称为虚拟现实、虚拟世界、灵境等,其基本特征可概括为:沉浸(Immersion)、交互(Interaction)和构想(Imagination)。它强调了信息处理过程中人的主导作用,使信息处理从以计算机为中心转到以人为中心。虚拟现实作为一项实用技术,目前广泛应用于产品原型设计、远程操作、教育、医疗、娱乐、训练等多个领域。

虽然独立的虚拟现实系统已在不少领域为用户带来了巨大的效益,但它在模拟现实世界时存在难以克服的复杂性。网络技术与虚拟现实技术相结合的产物—分布式虚拟环境可以更真实地模拟现实世界。分布式虚拟环境,又称为分布式虚拟现实、网络化虚拟现实、多用户虚拟环境等。它将分布在不同地理位置的独立的虚拟现实系统通过网络联结起来共享信息,多个用户在一个共享的三维环境中进行交互,协作完成一项任务,其中每个独立的虚拟现实系统称为一个“节点”或“主机”,每个用户在虚拟环境中用一个“实体(entity)”表示,也称为“化身(avatar)”或“对象(object)”,实体的状态由用户的输入来控制,并通过消息传递机制将“更新状态”在其他节点上反映出来。目前分布式虚拟环境已广泛应用于军事仿真训练、远程教学、远程会议、多用户游戏、产品异地设计等领域。

分布式虚拟环境<sup>[2]</sup>的研究最早可以追溯到1983

年 DARPA 和美国军方共同制定的 SIMNET 计划,到1990年,这一系统包括了约260个地面装甲车辆仿真器和飞机飞行模拟器,以及通讯网络、指挥所和数据处理设备,这些设备和人员分布在美国和德国的11个城市。通过这个系统可以训练军事人员和团组,也可以对武器系统的性能进行研究和评估。美国国防部与工业界在 SIMNET 成功的基础上,共同促进了分布交互仿真(Distributed Interactive Simulation,简称 DIS)技术的发展,形成了分布交互仿真的第一个标准—IEEE 1278 标准集。随后,分布交互仿真技术在军事仿真,尤其是集团训练中得到了广泛的应用。随着仿真规模的增加,美国国防部为了提高仿真的重用性和互操作性,支持大规模多兵种的仿真,在 DIS 的基础上开发了新一代的分布交互仿真体系结构 HLA 标准<sup>[3]</sup>,第一个基于 HLA 的分布式仿真系统 STOW97<sup>[4]</sup>的实体数目达到8000个,覆盖的地理区域有500\*700平方公里,而且战场环境更加丰富,仿真精度更高,增加了烟、雾、气象等各种环境效果,并开始支持动态地形环境,如弹坑、掩体、被毁的建筑物等。

组成一个分布式虚拟环境的基本部件有人机交互、逼真的三维虚拟环境、各种虚拟实体、网络通信以及将上述部件结合在一起的软件体系结构<sup>[6]</sup>。本文从这几个方面综述目前一些系统通常采用的技术。

## 2 人机交互

人机交互是指参与者通过各种交互设备(如数据头盔、数据手套、数据衣、触觉和力反馈设备等)与计算机相连,将各种动作反映到“化身”上,然后通过“化身”

\* )本文得到国家863高科技项目基金资助。赵沁平 教授,博导,主要研究领域为虚拟现实、类比推理。何红梅 博士生,主要研究领域为虚拟现实。

作用于虚拟环境或与虚拟环境中的其他对象交互。

若虚拟环境是对虚幻世界的模拟(如各种游戏),人机交互的设计目标是极大方便参与者与系统的交互;而若虚拟环境是对真实世界的模拟(如各种车辆、坦克、飞机仿真器),需要原样复制设备的位置、颜色、形状等,一方面使操作人员适应未来的真实环境,另一方面对未来仿真器的各种性能(如仪表位置、座椅的高低等)进行评估。

由于人对延迟非常敏感(小于100ms),所以实时性是人机交互的重要需求。产生延迟的因素有:各种输入设备延迟、绘制延迟、网络延迟等。这些都会降低仿真的真实度,如在车辆仿真中,当视景复杂时,帧速下降;而当视景简单时,帧速提高。这样,即便车辆是匀速运动的,也容易产生时快时慢的感觉。为了降低延迟,通常用多机系统分布式处理各种设备的输入,如用不同的工作站来处理头和手的输入。

### 3 逼真的三维虚拟场景

三维虚拟场景是虚拟环境的重要组成部分,它既可以是真实存在的某一地理区域的模拟,也可以是想象中的世界。它的逼真度是影响虚拟环境沉浸感的重要因素。逼真的三维虚拟场景的研究涉及到三维场景的建模、表示、管理和绘制等几个方面,下面主要讨论真实自然环境的三维场景建模和绘制。

#### 3.1 三维场景建模

真实自然环境在计算机中的三维表示,称为综合环境。综合环境数据库(也称为地形数据库、世界数据库)包括的要素有:①地形表面;②自然的和人造的特征物,总称为文化特征物,如道路、河流、桥梁、建筑物等;③气象,如雨、雾、阴、晴、白昼、夜晚等;④对象与环境交互产生的各种效果,如爆炸、火光等。综合环境数据库具体构造步骤如下:

(1)根据仿真应用的需要,选取合适精度的数字化高程数据、文化特征数据、航拍或卫星、遥感照片等多种地理数据。通常大多数地理数据来源于不同比例尺的地图。各类实体仿真时所需的地理数据的比例尺如表1。

表1 仿真实体类型与地理数据比例尺的关系

| 仿真实体类型 | 地理数据比例尺   |
|--------|-----------|
| 高空飞行仿真 | 1:250,000 |
| 低空飞行仿真 | 1:50,000  |
| 地面车辆仿真 | 1:10,000  |
| 单兵     | 1:5,000   |

目前常见的各种数据源包括:

①采用规则网格存储的地形数据。主要有下面几

种格式:

• USGS DEM(Digital Elevation Model):美国地质勘察局的数字化高度模型;

• NIMA DTED(Digital Terrain Elevation Data):美国国防部地图局的数字化地形高度数据。

②从地图或照片上抽取的采用向量格式存储的文化特征物数据,如河流、湖泊、道路、居民地等。主要有下面几种格式:

• NIMA DFAD(Digital Feature Analysis Data):美国国防部地图局的数字化特征分析数据;

• NIMA VPF(Vector Product Format):美国国防部地图局的向量产品格式;

• USGS DLG(Digital Line Graph):美国地质勘察局数字化线图;

• USGS LULC(Land Used and Land Cover):美国地质勘察局的土地使用图。

(2)对所收集的数据进行一致性处理,包括数据的筛选、简化、格式转换等,最终形成精度一致的高程数据与文化特征物数据,作为以后三维建模的基础。

(3)生成特征物的三维模型,如道路的加宽、湖泊的平放等。

(4)三维特征物模型与地形的整合,形成三角形不规则网格。具体有如下的两种方式:①直接将特征物贴在地形表面上;②在生成三维特征物模型后,再选取地形点来生成整合模型。

(5)在整合后的地形模型上加入其他的三维模型,如建筑物、树木等独立地物,最终形成综合环境数据库。

上述过程是静态综合环境的建模过程,而且主要描述的是三维地形环境的建模,目前有一些软件(如Multigen等)可以辅助这一建模过程。在三维地形环境的建模时还需考虑:①图形绘制的实时性与逼真性的矛盾,以便建立合理细节层次模型;②适合仿真实体需要的装载模块的划分。

除此之外,综合环境建模还应考虑到仿真过程中的地形环境变化以及各种仿真应用程序对它的需求,包括:①桥梁、房屋等静态对象的炸毁,需建模时建立多种毁伤模型;②各种环境效果,如雨、雾等各种气象及对象交互的各种效果,如爆炸等;③一些非可视化的信息,如各种地面材质等。

#### 3.2 三维场景绘制

三维场景绘制是将上述建模时生成的三维几何模型,采用透视投影的方法,将其绘制到计算机屏幕上去。它以参与者的视点来绘制三维场景,当视点移动时,场景会根据新视点的位置和方向重新绘制,如同在真实世界中一样。它所涉及到的核心是三维图形技术。

图形绘制要处理模型的几何信息(点、线、面)、光照、消隐、纹理映射、反混叠等,计算量会很大,一般采用图形生成硬件(Computer Image Generator,简称CIG,也称为图形加速器)。目前在高档图形工作站上(如SGI Onyx2),每秒可以绘制上百万个多边形,随着CIG价格的下降,在装有图形加速器的微机上也每秒可以绘制上万个多边形。

在三维图形绘制中,实时性和逼真度是一对矛盾,传统图形学的研究着重于图形的逼真度,采用光线跟踪、辐射度等技术。这样CIG系统需要处理的信息量较大,在低档平台上很难达到实时绘制。而在虚拟环境中,由于有人的交互参与,要求视点变化时,场景可以很快刷新。因此,对实时性的要求要高于逼真度。

图形绘制的实时性用帧速来衡量,而多边形的数目对帧速的影响最大,为了提高帧速(至少在15帧/秒以上),采用了多种技术,其中多细节层次(Level Of Detail,简称LOD)是广泛应用的一种技术。采用这种技术,当物体离视点较远时,用较少的多边形绘制;而当物体离视点较近时,用较多的多边形绘制。另外,微机平台上,图形绘制开始采用DirecX、Glide等标准,以牺牲逼真度,来提高实时性,如简化光源的计算等。随着CIG系统的发展,三维场景逼真度与实时性的矛盾会逐步解决。

为使参与者产生较强的沉浸感,在图形显示方面,根据人眼产生立体图像的原理,出现了多种立体显示装置,如头盔、立体眼镜、CAVE系统等。

#### 4 虚拟实体

在分布式虚拟环境中,仿真实体(如坦克、飞机等)可以由参与者直接操纵的,或者是由计算机自动生成并控制的。后者也称为计算机生成角色(Computer Generated Actors,简称CGA),其行为特征与人控制的近似,它可以丰富虚拟环境,而且降低系统开销。在虚拟战场环境中,CGA也称为计算机生成兵力(Computer Generated Forces,简称CGF),它们可以按照动力学方程运动,进行障碍物规避,而且还可以根据战场态势作出开火、躲避等实时反应。

在虚拟环境中将这些虚拟实体的各种特性表现出来,称为实体建模。通常包括:

- 几何特性:一般采用多边形表示的多面体。
- 物理特性<sup>[14]</sup>:质量、材料、密度等。如对于虚拟环境中的杯子,当在重力作用下摔到地板上,若是玻璃材料应当摔碎,是塑料制的,则不会。
- 行为特性:对象在特定环境中的动作。如在战场环境中坦克与树碰撞后,将树撞倒;导弹击中房屋后,将其摧毁等。图1是MR中对坦克行为的描述。

- 动力学特性:为某些仿真对象(如坦克、飞机),建立动力学模型,根据人机交互输入,或者计算机自动设定的参数,实时解算动力学方程,推算出动态对象的下一个位置,从而提高仿真的逼真度。

|                           |            |
|---------------------------|------------|
| instance Tank             | 坦克实例的行为描述  |
| actions;                  |            |
| collision(Tank)explosion; | 与其他坦克相撞后爆炸 |
| collision(Bomb)explosion; | 被炮弹击中后爆炸   |
| end                       |            |

图1 MR中的坦克行为描述

#### 5 网络通信

分布式虚拟环境中的网络通信保证了网上各节点时间和空间的一致性。下面从通信模式、数据分布模型、规模扩展几个方面进行讨论。

##### 5.1 通信模式

分布式虚拟环境的通信模式有对等模式(peer-to-peer)和客户-服务器(client-server)模式两种;消息传递方式可以是点播、广播或组播(如图2);采用的传输层协议可以是TCP或UDP。

对等模式允许网上任意两节点间直接通信,如SIMNET、NPSNET<sup>[1]</sup>、MR<sup>[5,12]</sup>、MASSIVE<sup>[3]</sup>等系统;客户-服务器模式则要求客户之间的通信必须通过服务器,即首先由某个客户将消息发送到服务器,然后再由服务器将其发送到其他客户,如BrickNet<sup>[10]</sup>、SPLINE<sup>[23]</sup>、RING<sup>[4]</sup>等系统。对等模式与客户-服务器模式相比,前者每个节点完全自治,没有哪个节点会成为瓶颈,或者因其崩溃而使系统瘫痪;而且消息传输延迟较小,后者则相反。但在客户-服务器模式中,服务器在向其他客户发送消息之前,可以对消息进行裁剪、增强或修改;而且可以将消息的处理负载从客户端移到服务器端,使客户端的通信负载下降;同时在服务器端加入新的服务也比较方便。至于分布式虚拟环境采用哪种模式,则与其应用的领域相关。为了克服单服务器的缺陷,几乎所有的客户-服务器系统如BrickNet、Spline、RING都考虑了多服务器,通常是对等模式和客户-服务器模式的结合。比如,在RING中,客户之间通过服务器通讯,服务器之间采用对等模式通信,而且服务器之间的通讯对客户来说是完全透明的。一般在这样的系统中,虚拟世界按照空间或其他关系被划分为一些子世界,每个服务器负责管理一个或多个子世界。

点播需要在任意两节点间建立联接。假定虚拟环境中有N个节点,则需建立 $O(N^2)$ 次联接。一旦某个实体的状态发生改变,同一条消息需发送 $(N-1)$ 次,而

广播则一条消息只需传递一次,但所有的节点均能收到并进行处理。因此,点播和广播对网络带宽的需求都很大,因此它们都很难支持系统的规模扩展,如在采用点播的MR中存有全拓扑图,支持5个以上的节点就很困难。在采用广播的系统SIMNET中,若有100,000个实体参与仿真,对网络带宽的要求为375M。组播(一般为IP Multicast)在广播的基础上有所提高,可以使一条消息一次发往网上部分节点,而不是所有节点,节点可以同时属于一个或多个组播组,通过它们只接收那些它所感兴趣的消息,因此,组播的使用可以大大降低网络负载和每个节点的计算负载。目前的大规模分布式虚拟环境如NPSNET、MASSIVE-2、STOW97中均采用了这项技术。使用组播技术的关键在于如何合理地建立组播组,即如何才能使消息发往真正需要它的那些节点,这一点将在5.3节中详细介绍。

传输层协议有可靠传输(如TCP或可靠的组播协议)和尽力传输(如UDP)两种。广播和组播通常都采用尽力传输协议,而点播可以采用可靠传输或尽力传输。可靠协议由于要等待应答信号而增加了消息传递的延迟,从而影响了系统的实时性和规模扩展;而尽力传输协议由于不必等待应答,可以很好地满足实时性,但在广域网上,会由于消息的丢失而无法保证消息的可靠传输,因而造成数据库的不一致。因此,有一些系统,如GMD<sup>[1]</sup>和基于HLA的分布交互仿真系统,既支持可靠也支持尽力消息传递。通常对于频繁更新的事件,如实体(化身)运动的位置、方向等,由于它的不可靠不会对系统构成很大影响,所以采用尽力传输协议;而对于不常发生的事件,如实体之间的碰撞,开火、爆炸等,由于无法重复,发生频率较低,采用可靠协议传输。

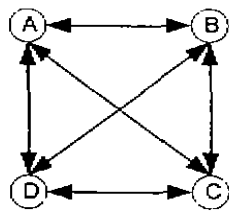


图2a 对等、点播

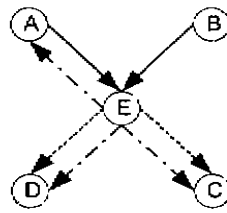


图2b 对等、组播

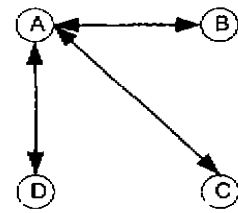


图2c 客户-服务器、点播

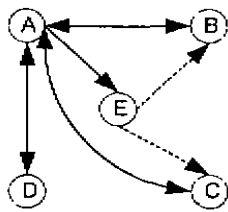


图2d 客户-服务器、组播

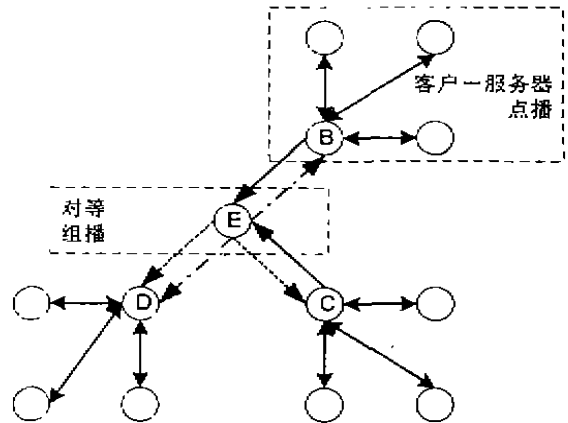


图2e 混合模式

图2 各种通信模式

## 5.2 数据分布模型

数据分布模型是指各节点共享的虚拟世界数据库在网上的发布方式和规律,以及保持分布数据一致性的机制。共享的虚拟世界数据库包括地形数据库、实体模型数据库和仿真过程中对象的当前状态等。从分布模式来看有分布式和集中式两种,而分布式中又有完全相同的分布和不完全相同的分布两种。不同的数据

分布模型会对分布式虚拟环境的规模、通信需求和数据的可靠性产生影响。

(1)集中式数据库 网上的多用户游戏MUD<sup>[1]</sup>一般采用共享的集中式数据库,由服务器维护唯一的世界数据库。每个客户节点将改变信息传输给服务器,然后由服务器修改数据库后,再将最新信息发布给其他客户节点,而且在某个时刻只有一个用户可以修改

数据库,这种模型对客户的数目有所限制,而且服务器很容易成为瓶颈。

(2)完全相同的分布式数据库 这一方法将完整的虚拟环境状态分布于网上的每个节点,由各节点自身维护数据库的一致性。通常的分布交互仿真系统采用这一模型。在仿真初始化时,每个节点拥有相同的虚拟世界数据库(主要是指地形环境),仿真过程中,网上只传输实体的状态,如位置、方向等,而且在每个节点上要维护虚拟世界中所有实体的当前状态。这一方式的优点是网上传输的消息的长度很小,缺点是结构不够灵活,如增加新的实体对象几乎是不可能的,而且世界数据库的改变会影响所有的节点。根据消息传输所采用的底层协议的不同(可靠或尽力),有近似相同和完全相同两种。

早期的 SIMNET 系统采用这样的模型(近似相同),随后几乎所有的分布交互仿真系统均采用了这一模型。但在大规模分布交互仿真系统中,每个节点在仿真过程中维护所有实体状态的计算开销较大,几乎不可能。因此有些系统对其进行了改进,如在 NPSNET 中,虽然每个节点可能仍然具有相同的地形数据库和模型数据库,但在仿真过程中,每个节点只调度和维护在其感兴趣区域中的地形环境和实体。

DIVE<sup>[1]</sup>系统将世界数据库划分为一些世界,世界实现为进程组,即一个世界是一个进程组,它采用可靠的组播协议传输消息。每个参与进程主动地管理各自的世界数据库副本,具有完全一致的数据库。

(3)不完全相同的分布式数据库 采用这一方法的系统,每个节点拥有不完全相同的数据库,其通信模式有客户-服务器结构和对等结构两种。

在 BrickNet 中,世界数据库在客户间进行划分,每个客户可以拥有不同的部分世界数据库,通信由服务器协调,每个客户的虚拟世界管理自身的对象集,其中部分或全部可以与网上其他节点的虚拟世界共享。服务器上存有每个客户的信息,当虚拟环境中某个对象的状态发生改变,共享此对象的其他虚拟世界将通过服务器的对象请求代理更新对象。BrickNet 非常适于大型的 CAD 系统,如可以将一个完整的设计分布于多个节点上,每个节点负责其中一部分的设计,并通过将其设计的对象与其他节点共享来协作完成较大的设计任务。

MR 采用对等通信模式,也不要求每个节点共享相同的对象集。每个节点可以管理自身的对象和实例集,其中部分或全部可以与网上其他节点共享。基于 HLA 的分布交互仿真系统也采用类似的方法,每个节点可以声明它所感兴趣的对象类型以及它可以与其他节点共享的对象类型。

### 5.3 规模扩展

几乎所有的分布式虚拟环境系统的设计均考虑到系统的规模扩展,这里的规模主要是指参与节点的数目,通常要解决两个核心问题:①如何使系统在节点数目增加时,网络的带宽需求不会太高;②如何使系统在节点数目增加时,各节点的处理负载不会太高,即不会有网络阻塞或不会有太多的共享对象处理。为此,各系统提出了不同的方法,如在 SIMNET 中提出的递推算法可以较大地降低网络的带宽需求,但主机的计算负载仍然很大。

目前在大规模分布式虚拟环境中,为解决上述的两个问题,大多采用组播技术,将组播组(即组播地址)与一种功能或虚拟世界的某个范围相关。由于组播技术一方面可以使信息只在那些真正需要它的节点之间共享,而且一次可以将一条消息发往多个节点,因此对网络带宽的需求不会太高。另一方面由于每个节点接收的消息有限,从而处理的实体数目也是有限的,因此各节点的计算负载也会下降。

在分布式虚拟环境中,使用组播最简单和最直接的方法是为整个世界分配一个组播组地址并用它将消息发送到所有的参与者。它要求世界的每个参与者接收并处理每条消息,无论消息是否与接收者相关(比如对象之间的距离可能很远相互根本看不到也听不到)。SIMNET 和早期的 NPSNET 中使用了这一方法。另一种情形是,为虚拟环境的每个对象的每种媒体分配一个或多个组播组。这样,在具有大量对象的虚拟环境中,需要许多的组播组,而每个组只处理很小的带宽。

每个组播组都需要在支持组播的路由器、内核和协议栈中进行管理,费用较高,而且组播组也是有限的,所以组播组地址不能滥用。因此,需要一种折衷方案,使一些相关的对象可以共享一个组播组,在保证每个组的带宽尽量小的情况下,降低所需的组播组的数目。下面是目前各种系统采用的一些方法:

NPSNET-IV 将虚拟环境划分为一些正六边形单元(cell),每个单元与一个组播组地址相关,而且每个单元与组播地址的映射关系是在仿真之前就建立好的。每个观察者都有各自的感兴趣区域(Area Of Interest,简称 AOI),一个感兴趣区域可以覆盖多个单元。当观察者(如由人操纵的仿真器)运动时,可以动态地加入、退出多个组播组。根据实验测试,这种划分可以支持到 50000 个实体,非常适于战场环境。但在战场环境范围较大时,这种统一的组播组的划分可能会用光所有的组播组。

Spline 将虚拟世界划分为一些不相连的任意形状的区域,称为 Locales。每个区域对应一个组播地址,可以是动态分配的。每个区域都存储与其相邻的其他区

域的链表,对象可以从一个 Locale 方便地移动到另一个 Locale, GMD 也采用了类似的不规则“zoning”的划分,但其组播地址是静态分配的。

MASSIVE-2 提出了一种基于“第三方对象(Third Party Object)”的通用世界划分方法。第三方对象是独立于虚拟世界其他实体的抽象概念,如建筑物中的多个屋子,每个屋子可以看作一个第三方对象;距离观察者很远或观察者不处于其中的一群人也可以看作一个第三方对象;第三方对象可以是抽象的,也可以有具体的可视表示,如一群人的视觉表示。它可以动态地建立、删除。实现时,每个第三方对象用一个组对象(Group Object)表示,每个组对象管理一个或多个组播组。

HLA 提出了路由空间(Routing Space,简称 RS)的概念。RS 是个多维坐标空间,每个盟员(即网上的一种应用程序)可以根据路由空间定义更新区域和预购区域,将某条消息发往某个更新区域或只接收某个预购区域中的消息,而且区域也可以修改。如观察者(如人操纵的坦克仿真器)只能看到 10km 范围内的视界,其预购区域便在这个范围内,而随着坦克的运动,其更新和预购区域也在不断变化。当某些盟员的更新区域和预购区域重叠(相交)时,便在这些盟员间建立一条 Multicast 的链路(即使用同一个组播组收发消息)。路由空间的定义与相应的应用相关。如在 STOW97 中定义的路由空间有地理区域、无线电波段等。

## 6 软件体系结构

软件体系结构将上述的网络管理、图形绘制、人机交互、碰撞检测等多个部件合理而有效地结合在一起,形成一个系统。通常采用的方法是把这些不同的功能划分为一些并行单元,通过某种方式共享数据。这样,一方面提高了系统的性能(如可以将并行单元分布于多处理器上计算),另一方面添加新功能也比较容易,下面是目前广泛使用的平台或应用系统采用的一些方法。

### 6.1 MR

MR (Minimal Reality)<sup>[9,13]</sup>工具箱是由加拿大 Albert 大学图形学实验室开发,支持单用户和网络虚拟现实系统开发的软件工具包。MR 可以分为三层:底层的设备支持,数据处理和高层的服务(如图 3)。

MR Toolkit 的软件体系结构采用“分解的仿真模型”,将虚拟现实系统划分为表示、交互、几何模型和计算四个部分(如图 4)。其中计算部件是应用程序的核心,与其他系统部件的处理相独立并为几何模型部件生成数据;几何模型部件在计算中保持数据的高层表示并将数据转换为适于表示部件处理的格式;表示部件将应用数据表现出来以使用户可以看到、感觉到;交

互部件管理所有的用户输入,同时将输出协调后呈现给用户。上述四个部件可以分布在网上多台工作站上处理。每个基于 MR 的应用程序中都有一个主进程(master),它启动应用程序中的其他从进程(slave)或计算进程(computation)的运行。master 与 slave 通过建立的 socket 共享数据。

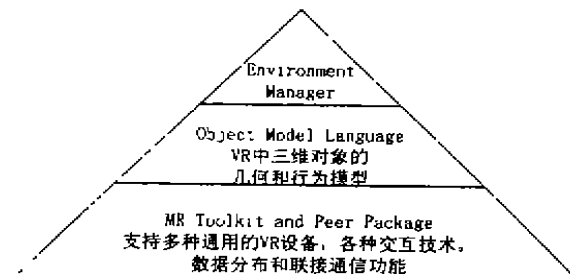


图 3 MR Toolkit 的组成部分

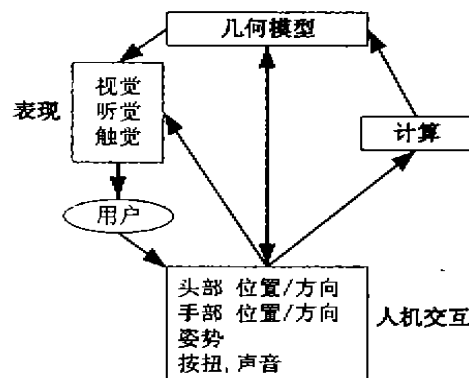


图 4 MR 分解模型中的四个部件

Peer Package 是对 MR Toolkit 的扩展,提供了应用程序主进程之间的消息收发机制,而从进程之间的通信必须通过主进程。

对象建模语言 OML 描述了独立于平台的对象的几何特性和行为特性,采用 OML 编译器可以生成一种对象中间表示,并由嵌入应用程序中的 OML 解释器执行。

环境管理器 EM 是为降低虚拟环境系统开发的复杂度而设计的。EM 通过解析描述文件来初始化虚拟环境,支持单用户和多用户的虚拟环境,并提供了监控虚拟环境运行的功能。描述文件中包括的内容有:设备配置、OML 的对象列表、OML 的行为列表、OML 的对象实例列表以及在多用户虚拟环境中共享的对象和实例信息等。

### 6.2 NPSNET-IV

NPSNET-IV<sup>[6]</sup>是由美国海军研究生院计算机系开发的用于军事训练和仿真的分布式虚拟现实系统。

它是第一个采用 IP-Multicast 来支持大规模仿真的分布式虚拟现实应用系统。

在 NPSNET-IV 中,每个实体在一台特定的主机上仿真,实体间采用 DIS 协议进行通信。每台主机可以仿真一些对象并采用递推(DR)模型来仿真其他非本地控制的实体,当一个本地实体的状态与其相应的 DR 状态比较超过某一阈值时,便将当前状态发向网上其他节点。

NPSNET-IV 运行于 SGI Onyx2 工作站上,其软件体系结构(如图 5)充分利用了具有 Reality Engine2 图形生成器的 SGI Onyx2 的多处理器结构和其上 Performer 图形库提供的并行性。

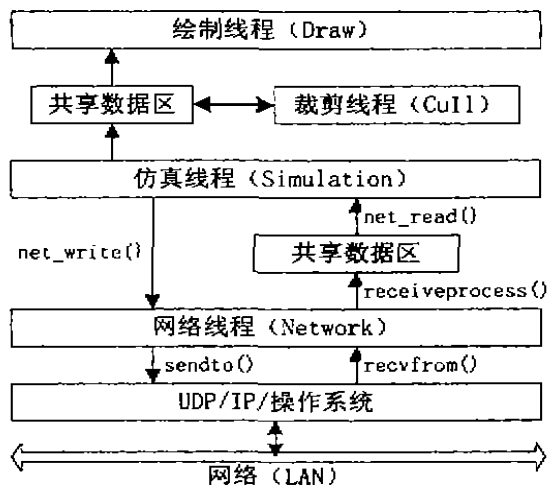


图 5 NPSNET 软件体系结构

为了实现高性能图形效果,Performer 库支持多线程存取,其中包括裁剪(Cull)和绘制(Draw)线程。NPSNET-IV 继承了这两个方法。在 Cull 线程中遍历世界数据库,选择那些在视锥中的几何对象,并选择其细节层次,为 Draw 线程构造显示列表。然后 Draw 线程绘制显示列表中的内容。NPSNET-IV 采用四处理器,图 5 中的四个线程 Draw、Cull、Simulation 和 Network 分布在四个处理器上进行计算。

#### 参考文献

- 1 Broll W. Bringing People Together — An Infrastructure for Shared Virtual Worlds on the Internet. In: Proc. Sixth IEEE Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (June 18 — 29, 1997, MIT, Cambridge, Massachusetts), Los Alamitos, California: IEEE Computer Society Press, 1997. 199~204

- 2 Budge L D Synthetic Theater War (STOW) 97 Overview. 1998 Spring Workshop on Simulation Interoperability. Available at: <http://siso.ist.ucf.edu/stw/98Spring/>
- 3 Carlsson C, Hagsand O DIVE — A Platform for Multi-User Virtual Environments. Computer & Graphics, 1993, 17(6): 663~669
- 4 Funkhouser T A. RING: A Client-Server System for Multi-User Virtual Environments. ACM SIGGRAPH Special Issue on 1995 Symposium on Interactive 3D Graphics, (Monterey, CA), 1995. 85~92
- 5 Greenhalgh C, Benford S. Supporting Rich and Dynamic Communication in Large-Scale Collaborative Virtual Environments. Teleoperator and Virtual Environments, 1999, 8(1): 14~35
- 6 何红梅. 分布式虚拟战场环境系统结构的研究与实现: [博士论文]. 北京航空航天大学, 2000
- 7 IEEE P1516 Draft Standard for Modeling and Simulation (M&S) High Level Architecture (HLA), Version, 1. 3, April 1998 Available at: <http://hla.dmsomil/>
- 8 Macedonia M R, Zyda M J, et al. NPSNET: A Network Software Architecture for Large Scale Virtual Environments. Teleoperators and Virtual Environments, 1994, 3(4)
- 9 Shaw C, Green M. The MR Toolkit Peers Package and Experiment. In: IEEE Virtual Reality Annual International Symposium (VRAIS'93). 1993. 463~469
- 10 Singh G, Serra L, Png W, Wg H. BrickNet: A Software Toolkit for Network-Base Virtual Worlds. Teleoperators and Virtual Environments, 1994, 3(1): 19~34
- 11 Towell J, Towell E. Presence in Text-Based Networked Virtual Environments or "MUDs". Teleoperators and Virtual Environments, 1997, 6(5): 590~595
- 12 汪成为, 高文, 王行仁. 灵境(虚拟现实)技术的理论、实现及应用. 清华大学出版社, 1996
- 13 Wang Q J, Green M, Shaw C. EM — An Environment Manager for Building Networked Virtual Reality, VRAIS'95
- 14 王兆其. 虚拟环境中物体运动逼真性的研究: [博士论文]. 北京航空航天大学, 1999
- 15 Waters R, et al. Diamond Park and Spline: A Social Virtual Reality System with 3D Animation, Spoken Interaction, and Runtime Modifiability: [TR-96-02a]. Available at: <http://www.merl.com/>
- 16 赵沁平, 沈旭昆, 夏春和, 王兆其. DVENET: 一个分布式虚拟环境. 计算机研究与发展, 1998, 35(12): 1064~1068