

通用分布式虚拟现实软件开发平台的研究

邓志东, 余士良, 程振波

(清华大学 计算机系, 智能技术与系统国家重点实验室, 北京 100084)



摘要: 为了适应分布式虚拟现实应用系统的开发需要, 在通用虚拟现实软件开发平台 VRMagic 的基础上, 结合高层体系结构(HLA), 研发出通用分布式虚拟现实软件开发平台 D-VRMagic。详细介绍了 D-VRMagic 的网络架构和对象模型的构建过程, 并在此基础上提出了管理者设计规范和通用客户端模型。最后以网络五子棋的实现为例, 对 D-VRMagic 进行了验证。结果表明该平台具有较高的开发效率、可扩展性和通用性。

关键词: 分布式虚拟现实; 高层体系结构; 通用开发平台; 网络架构; 对象模型

中图分类号: TP 391.9 **文献标识码:** A **文章编号:** 1004-731X (2008) 12-3160-05

General-purpose Distributed Virtual Reality Development Platform Based on High Level Architecture

DENG Zhi-dong, YU Shi-liang, CHENG Zhen-bo

(National Laboratory of Information Science and Technology, Department of Computer Science, Tsinghua University, Beijing 100084, China)

Abstract: In order to meet the demand for rapidly developing distributed virtual reality application, a general-purpose distributed virtual reality development platform, i.e., D-VRMagic, based on our general-purpose development platform for virtual reality (VRMagic), was developed. The D-VRMagic was built with the framework of the High Level Architecture (HLA). First, the network architecture and the object model of the D-VRMagic were described. Second, the design specification of manager federate and the model of the common client were given. Finally, a networked gobang game was designed and implemented using the D-VRMagic. The experimental results show that the new generation platform has a high efficiency, scalability, and universality.

Key words: DVR; HLA; general-purpose development platform; network architecture; object model

引言

分布式虚拟现实技术^[1] (Distributed Virtual Reality, DVR) 又称为网络虚拟现实 (Networked Virtual Reality, NVR), 是虚拟现实技术与计算机网络技术相结合的产物。它通过计算机网络, 使分散的用户参与到同一虚拟场景, 并在其中进行自然的交互。目前, 分布式虚拟现实技术已经广泛地应用于军事^[2]、医学^[3]、航空^[11]、制造业^[14]、教育、娱乐等领域。

分布式虚拟现实系统涉及网络体系结构、通讯和管理、软件体系结构、共享场景的时空一致性、视景的建模、管理和生成等, 开发难度大, 而且效率低。改善分布式虚拟现实系统开发环境, 提高开发效率, 成为分布式虚拟现实技术发展迫切需要解决的主要问题之一。目前, 国内外的一些企业和科研单位对分布式虚拟现实软件开发平台进行了一些研究, 如加拿大 Alberta 大学的 MR Toolkits^[4]、美国 Sense8 公司的 World Tool Kit (WTK)^[5] 和国内北京航空航天大学 DVENET^[6] 等。这些平台在一定程度上有助于分布式虚拟现

实系统的开发, 但其操作难度高, 开发效率较低。

我们前期已开发出一套通用虚拟现实软件开发平台 VRMagic^[7]。该平台能够帮助用户快速构建虚拟现实应用系统, 而且操作简便、开发效率高、通用性强。但是, 它并不支持分布式应用。为了解决这个问题, 我们在原有的 VRMagic 基础之上, 通过引入 HLA 技术^[8,9,12,13], 使各个分散的节点实现互联, 从而开发出了一种分布式通用虚拟现实软件开发平台——D-VRMagic。新一代 D-VRMagic 平台不仅保留了 VRMagic 的全部优点, 而且由于引入了 HLA 的可重用性与互操作性, 因而具有很强的可扩展性、灵活性与通用性。D-VRMagic 同时具有六自由度漫游、实时在线交互、实时视频合成与 GIF 动画等功能, 并且支持立体眼镜、V8 头盔、CyberGlove 数据手套与多种定位与跟踪设备。

本文首先描述了 D-VRMagic 的网络架构和对象模型, 然后给出了管理者联邦成员的设计规范和通用客户端模型。最后以多玩家五子棋游戏为例, 介绍了如何基于 D-VRMagic 快速完成分布式虚拟现实应用系统的开发。

1 D-VRMagic 的总体设计

1.1 网络架构设计

D-VRMagic 遵循 HLA 规范。在 HLA 规范中, HLA 将整个系统视为一个联邦 (Federation), 系统中的每个节点称

收稿时间: 2007-03-16

修回日期: 2007-06-01

基金项目: 国家自然科学基金 (60621062, 60775040)

作者简介: 邓志东(1966-), 男, 四川宜宾人, 教授, 博导, 研究方向为虚拟现实技术、计算智能、复杂网络理论、计算生物学等; 余士良(1982-), 男, 福建漳州人, 硕士生, 研究方向为分布式虚拟现实; 程振波(1975-), 男, 江西鄱阳县人, 博士生, 研究方向为计算智能与计算神经科学。

为联邦成员(Federate)。联邦成员通过运行时间支撑环境(Run-time Infrastructure, RTI)联系在一起, 相互协作完成预定的任务。HLA 的联邦规则明确规定了联邦执行范围内的信息交换都必须通过 RTI 来实现。RTI 是 HLA 接口规范的软件实现, 封装了仿真运行管理和底层通讯。因此, D-VRMagic 不需考虑各种复杂的网络环境, 只要选用合适的 RTI 中间件即可。在 D-VRMagic 的设计中, 我们选择了美国国防部资助开发的 DMSO RTI^[10]。

根据功能划分, 本文为 D-VRMagic 设计了两类联邦成员, 分别是客户端联邦成员(下文简称客户端)和管理者联邦成员(下文简称管理者)。其中, 客户端联邦成员运行在用户 PC 上, 是系统与用户交互的接口。一方面, 客户端通过 RTI 接收虚拟场景信息并及时显示给用户, 保证本地的场景与系统场景相一致; 另一方面, 它接收用户通过交互设备, 如鼠标、数据手套等输入的操作指令, 并由 RTI 把场景的修改信息转发给其他联邦成员。管理者联邦成员运行在后台的服务器上, 提供与具体应用相关的服务, 包括客户端的管理、场景的生成和维护、用户操作行为的监控。

D-VRMagic 采用多管理者多客户端的网络架构, 如图 1 所示。多管理者通过 RTI 提供的所有权管理、数据分发管理等服务共同作用, 承担分布式虚拟现实系统由于复杂场景的生成与维护导致的大量计算开销。例如, 通过所有权管理, 允许不同的管理者维护不同对象的状态; 通过数据分发管理, 允许不同的管理者维护不同区域的场景。该网络架构能够充分发挥 RTI 的“即插即用”的优点, 具有良好的可扩展性。

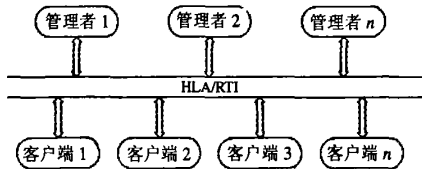


图 1 网络架构图

1.2 仿真描述文件的定义

通用仿真描述文件, 即 SAT 文件可对虚拟现实场景进行完整的定义。SAT 文件包括 SAT 头和 SAT 体两部分。其中 SAT 头定义了视点初始位姿、场景数、背景物体及其位姿、光源及其位姿、物体(对象组)及其位姿、关节定义、关节的初始位姿等, SAT 体定义了指令流和数据流。SAT 头格式如表 1 所示, SAT 体中指令流格式如表 2 所示, 数据流格式如表 3 所示。在表 1-3 中, s 表示字符串, i 表示整型数, f 表示浮点数, T 表示标识码, C 表示操作码。

1.3 对象模型设计

为了保证对象模型的通用性, 我们采用了面向对象的方法对 D-VRMagic 的共享数据进行了抽象, 设计了如图 2 所示的对象模型。该对象模型包括 5 个对象类(World、User、SATHead、SATBody 和 Node)和 1 个交互类(OPCorrect)。

表 1 SAT 头	
语句格式	说明
f	保留字段
i	保留字段
ffffff	视点位姿: 位置[3]+姿态[3]
i	场景个数
Tis	"Sc"+场景序号+场景名称
sffffff	背景物体名称+位置+姿态+缩放
i	光源的个数(N1)
N1	0(环境光)+强度+颜色
	1(点光源)+强度+位置
	2(聚光灯)+强度+位置+方向+衰减系数+角度
	3(有向光)+强度+方向
i	对象组个数(N2)
N2	0(无关节型)+名称+位置+方向+缩放
	1(有关节型)+名称+位置+方向+缩放
	2(工具类型)+名称+位置+方向+缩放
i	关节个数(N3)
N3	0(滑动关节)+父关节+子关节+起点+终点+支点
	1(旋转关节)+父关节+子关节+起点+终点+指点
ffff.....(N3个f)	各个关节的 Home 位移量
Si	运行开关(0: 停止 非 0: 运行)

表 2 指令流格式	
语句格式	说明
Cis[参数集]	C 帧序号 操作码 参数(因操作而异)

表 3 数据流格式	
语句格式	说明
Diffff.....(N3个f)	D 帧序号 N 个关节的相对位移

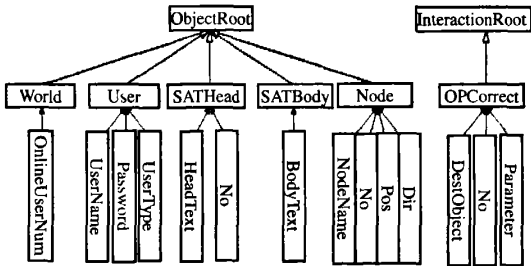


图 2 D-VRMagic 的对象模型

- 1) World 对象类用于管理者向客户端广播在线用户数。
- 2) User 对象类用于用户身份验证。该类包括用户名、密码和用户类型三个属性。客户端通过用户名和密码向管理者提交用户验证信息, 管理者通过用户类型向客户端反馈验证结果。
- 3) SATHead 对象类用于管理者向客户端发布仿真描述文件的 SAT 头。No 属性用于标记 SAT 头的生成时刻。
- 4) SATBody 对象类用于管理者向客户端发布 SAT 体。
- 5) Node 对象类用于发布场景中可操作物体的状态。其中 NodeName 类属性用于标记 Node 对象实例所对应的物体。Pos 和 Dir 分别表示物体的位置与姿态信息, No 表示更新序号。
- 6) OPCorrect 交互类用于管理者进行物体操作时的冲突消解。OPCorrect 包括三个参数, 其中 DestObject 表示修改

后的目标物体, No 表示修改后该物体对应于 Node 实例中的 No 属性值, Parameter 表示修改后物体的状态值。

1.4 管理者的设计规范

为了使基于 D-VRMagic 的应用系统能够直接使用 D-VRMagic 的客户端, 从而减少应用系统的开发任务, 管理者的设计须遵循如下七条规范:

- 1) 管理者负责联邦执行的创建和初始化;
- 2) 联邦范围内有惟一的 World 对象实例, 管理者负责 World 对象实例的生成和属性的更新;
- 3) 联邦范围内有惟一的 SATHead 对象实例, 管理者负责 SATHead 对象实例的生成和属性的更新, 其中属性的更新发生在其它联邦成员请求更新时;
- 4) 联邦范围内有惟一的 SATBody 对象实例, 管理者负责 SATBody 对象实例的生成和属性的更新。当其它联邦成员请求 SATHead 头或播放动画过程中, 管理者联邦成员将更新 SATBody 对象实例的属性值;
- 5) 管理者拥有联邦范围内所有 User 对象实例 Type 属性的所有权并负责其更新。当管理者反射 User 对象实例的 UserName 和 Password 属性值时, 将激活对该 User 对象实例的验证例程, 并根据验证的结果设置 Type 属性值。Type 属性实例的更新发生在管理者启动验证模块或其它联邦成员请求更新时;
- 6) 管理者负责 Node 对象实例的生成, 并一直保持对 NodeName 属性拥有所有权。Node 对象实例在创建伊始, 其所有权即为管理者所拥有。当其它联邦成员请求获取非 NodeName 属性实例所有权时, 管理者应放弃相应属性的所有权。而其它联邦成员放弃获取的属性实例所有权时, 管理者应及时回收相应属性的所有权。在放弃属性实例所有权期间, 管理者必须反射 Node 对象实例的更改。管理者在其他联邦成员请求更新 Node 属性值时, 对其所拥有的属性值进行更新;

7) 管理者在发现物体操作存在冲突时, 将通过发送 OPCorrect 交互实例进行冲突消解。管理者必须通过设置适当的 OPCorrect 实例的 No 参数值, 使交互实例对物体状态的修正具有最高的优先级。

在遵循上述规范的前提下, 应用系统开发人员可以根据具体应用的需求, 设计多个管理者联邦成员和它们之间的协作关系, 以制定用户在联邦执行时的行为。

2 D-VRMagic 客户端的设计与实现

2.1 客户端设计

在遵循统一对象模型的基础上, 我们把客户端分为用户界面(User Interface, UI)和通讯两个模块, 如图 3 所示。其中 UI 模块负责处理人机交互, 包括场景维护与显示、用户输入输出处理等; 通讯模块负责与 RTI 交互, 调用 RTI 接口和

处理 RTI 的回调。由于 DMSO RTI 不允许并发调用, 并且 RTI 的回调函数须由 tick(DMSO RTI 的函数)驱动, 因此 UI 模块和通讯模块不能共享一个线程。此时, UI 模块由程序主线程驱动, 通讯模块由另一个独立的线程——通讯线程驱动。其中, 通讯线程在客户端加入联邦后由程序主线程动态创建, 并在退出联邦后销毁。

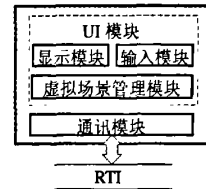


图 3 客户端程序架构

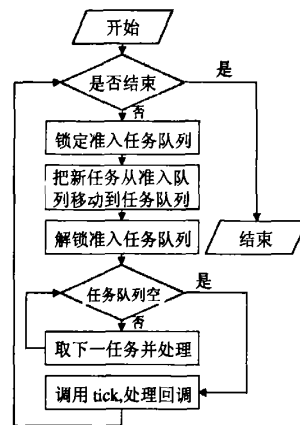


图 4 通讯线程流程图

程序主线程由 VC 自动生成, 在初始化之后循环 Windows 消息, 直至程序退出。通讯线程的流程图如图 4 所示, 循环处理任务队列中的任务体, 直至线程结束。任务体是 D-VRMagic 自定义的一种数据结构, 用于表示对类的非静态成员函数的调用或对类的静态成员函数的调用。UI 模块对通讯模块接口的调用, 可以分为同步调用和异步调用。在同步调用中, 由程序主线程直接执行通讯模块的相关代码。异步调用由通讯线程根据程序主线程提交的任务体, 异步地驱动相关代码的执行。在异步调用中, 程序主线程首先根据调用的对象生成任务体, 并把任务体加入到通讯线程的准入任务队列(客户端采用标准模板库 stl 的 map 来维护任务队列, 由于 map 并非线程安全, 因此可通过引入准入任务队列预先存放程序主线程提交的任务体, 来提高程序的并发度)。然后由通讯线程从准入队列中取出任务体予以执行。利用这种机制, 一方面可以避免由于通讯阻塞导致的程序界面停顿, 另一方面通过允许通讯线程对待处理任务进行合并处理, 如对同一对象实例的多次更新的任务体可以合并成一次, 相应可减少 RTI 的调用, 提高通讯的效率。通讯线程通过 Windows 消息的方式与程序主线程进行通讯。

从功能的角度来看,客户端包括初始化、登陆验证、场景显示、动画播放、退出系统、操作物体等功能模块,如图 5 所示。其中初始化模块包括程序的初始化、联邦的加入、HLA 句柄的获取、通讯线程的启动等功能。登陆验证模块把用户输入的用户名和密码提交给管理者,并根据管理者的验证结果决定是继续加入系统还是退出系统。退出系统模块包括退出联邦和释放系统内存等。场景显示、动画播放、操作物体等模块只有在用户成功登陆之后方被激活,每个逻辑模块都在 UI 模块与通讯模块的共同作用下实现。

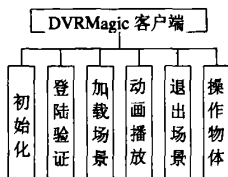


图 5 客户端功能模块图

2.2 功能模块设计

下面主要介绍场景加载、动画播放和操作物体这三个主要模块的设计。

2.2.1 场景加载

场景加载模块是客户端联邦成员通过身份验证后首先激活的模块。该模块包括预定 SATHead 对象类、发现 SATHead 对象实例、反射 SATHead 对象实例属性、场景生成和取消 SATHead 对象类预定等。其中场景生成是根据接收的 SAT 头载入相应的三维物体并显示在终端上。

2.2.2 动画播放

动画播放模块可保持本地场景与系统场景的一致。而导致系统场景改变的原因有三：一是管理者正在执行 SAT 体，二是客户端正在操作物体，三是管理者正在解决操作冲突。因此，系统将分别设置 SAT 体的接收模块、移动物体的观察模块和冲突消解模块。它们的功能分别是反射 SATBody、反射 Node 和接收 OPCorrect。

2.2.3 操作物体

物体的操作应包括三个步骤，即抓取物体、移动物体和放弃物体。为了提高程序的模块化程度，系统设计了三个子模块来分别处理以上三个物体的操作步骤。

用户在对物体操作之前必须通过抓取物体来获取物体相应 Node 的所有权，并以此保证物体在同一时刻只为一个用户所操作。当用户获取物体所有权之后，开始对物体进行移动，客户端周期采样物体的位置和姿态，并通过相应的 Node 对象通知其它的联邦成员。当用户操作物体结束之后，则立即放弃该物体的所有权，允许其它用户对其进行操作。

3 应用实例：五子棋网络游戏

本文以开发五子棋网络游戏为例，说明如何利用 D-

VRMagic 开发实际的分布式虚拟现实应用系统。该游戏设置如下：一个房间，一个棋盘、120 个白棋子和 120 个黑棋子，允许用户对任意棋子进行操作。

3.1 场景设计

场景设计包括三维物体建模、世界场景布局和 SAT 文件生成。利用 VRMagic 构造满足需求的三维虚拟场景，如图 6 所示。图中棋子按照黑白两种颜色分别叠放在同一个位置。

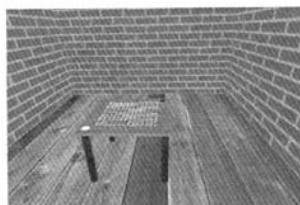


图 6 五子棋场景图

3.2 管理者设计

首先确定管理者个数，然后根据管理者设计规范设计并在 Visual C++ 上实现相应的功能。为了简化设计，我们只设置一个管理者。如图 7 所示，该管理者采用与客户端相类似程序架构，分为上层的管理模块和底层的通讯模块。管理模块由程序主线程驱动，通讯模块由独立的通讯线程驱动。程序主线程通过任务体异步地调用通讯模块的接口，通讯线程通过 Windows 消息通知程序主线程处理接收的数据或任务体的处理结果。管理模块负责游戏初始化、用户验证、SATHead 发布、场景维护、Node 所有权管理和冲突消解等。

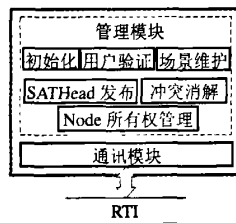


图 7 五子棋管理者联邦成员程序架构

3.3 功能模块设计

Node 所有权管理和冲突消解是管理者设计的难点和重点，这里我们主要介绍这两个模块的实现。

3.3.1 Node 所有权管理

根据 D-VRMagic 管理者联邦设计规范第 6 条，我们主要借助 RTI 的 attributeOwnershipAcquisition 和 negotiated-AttributeOwnershipDivestiture 这两个接口来实现 Node 的所有权管理。

在初始状态，管理者创建 Node 对象实例并拥有其所有权。如图 8 所示，首先通讯主线程通过调用 negotiated-AttributeOwnershipDivestiture 请求放弃所有权。当有客户端请求获取该 Node 实例所有权时，RTI 向通讯模块回调



图 8 Node 所有权管理序列图

attribute-OwnershipDivestitureNotification 函数，通知管理者放弃所有权成功，至此，管理者不再拥有 Node 实例的所有权。然后，管理者马上通过 attributeOwnershipAcquisition 请求获取 Node 实例的所有权。由于 RTI 不允许在回调函数中调用 RTI 的接口，所以，回调函数把该接口调用封装成任务体，放入通讯线程的准入任务队列，由通讯线程在处理任务体时再加以执行。当客户端无条件放弃 Node 实例的所有权之后，RTI 通过回掉函数 attributeOwnership-Acquisition-Notification 通知管理者又重新获取 Node 实例的所有权。然后，管理者马上重新请求放弃所有权，其处理方式与请求获取所有权一样，通过任务体加以实现。

3.3.2 冲突消解

D-VRMagic 管理者设计规范第 7 条规定由管理者负责冲突消解。在该游戏中，可能导致的冲突是不同的用户同时把棋子移动到同一个地方。

如图 9 所示，当通讯模块通过 RTI 的回调函数 reflect-AttributeValues 更新 Node 实例时，触发冲突检测。如何发现冲突，则管理者把 Node 实例对应的物体按照移动的轨迹退回到与碰撞物体刚好接触的位置，并通过发送 OPCorrect 交互实例转发给其它用户。为了保证该修正具有较高的优先级，管理者将 Node 实例的 No 属性分为高字节部分和低字节部分，并对高字节部分加 1。由于客户端在操作物体时，每更新 Node 实例对它 No 属性值的低字节部分加 1。所以管理者对高字节部分加 1，能够使接收方把操作者在修正时刻之后对物体的操作进行过滤，从而可相应提高游戏的逼真度。

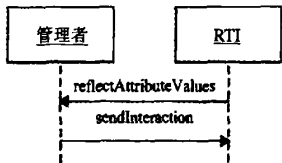


图 9 冲突消解序列图

3.3 运行结果与测试分析

我们在 10/100M 的局域网环境下对五子棋游戏进行了运行测试。测试环境包括五台 Pentium(R) IV 3.00GHz 计算机，其中一台机器运行 RTI 执行进程 RtiExec 和联邦执行进

程 FedExec，一台机器运行管理者联邦成员，三台机器运行客户端联邦成员，它们之间通过 10/100M 交换机互联，如图 10 所示。由于该测试用例没有涉及复杂的场景计算和大量的用户并发访问，五台机器的硬件配置如表 4 所示。操作系统均为 Microsoft Windows XP Professional Service Pace 2。

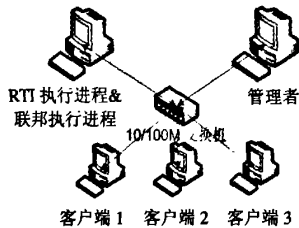


图 10 网络配置

表 4 计算机配置

硬件	配置
CPU	Pentium(R) IV 3.00GHz
内存	256Mx2 (DDR400)
显卡	GeForce FX 5700(128M 显存)
网卡	Realtek RTL8139 10/100M

管理者程序界面如图 11 所示。客户端界面如图 12 所示，其中左、右两幅图分别表示两个不同客户端联邦成员在同一时刻的程序界面图。



图 11 管理者联邦成员界面

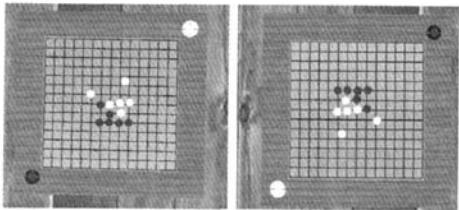


图 12 客户端联邦成员界面

游戏的性能测试结果如表 5 所示。

表 5 五子棋游戏性能测试结果

指标	测试结果
场景总面片数	26,868 个
本地操作延迟	<0.2 秒
异地操作延迟	<0.3 秒
客户端 CPU 占用率[空闲]	1%
客户端 CPU 占用率[漫游/操作物体]	50%
内存大小	51.6 兆
网络带宽使用率	<0.1%

(下转第 3186 页)

尖初始位置在 0 点, 则首先工件转过 A1 角, 同时刀具转过 B1 角, 然后工件转过(A2-A1)角, 同时刀具转过(B2-B1)角。

3.3.2 编制加工 NC 代码

为了使机床的加工程序编制具有通用性, 采用 ISO 标准规定的代码格式进行程序的编制, 再利用 3.3.1 所述方法实现 G00, G01, G02 等, 将 NC 代码导入虚拟数控机床系统, 即可完成仿真加工。NC 代码如下:

```
N20 G18 G00 X0.0 Z150.0;  
N25 S1200 M03;  
N30 G17 X-9.0 Y54.0;  
... ..  
N240 G01 X35 Y-21;  
N245 M05;  
N250 G40 G00 Z300.0;  
N250 M30;
```

3.3.3 虚拟加工过程仿真的实现

在用户界面中输入相关加工信息, 并启动虚拟加工仿真过程, 系统自动打开 AutoCAD 仿真平台中的机床实体模型, 根据 NC 代码翻译后获得的 G 指令和坐标值进行虚拟加工过程的实时仿真。加工仿真结果如图 5 所示。

4 结论

针对目前仿真软件大多开发难度大、周期长、通用性不足以及价格昂贵, 系统资源占有量惊人, 硬件配置要求高的情况, 本文利用 AutoCAD 的二次开发工具 ObjectARX 建立了 VisualC++6.0 与 AutoCAD 之间的通讯, 将 VisualC++6.0 作为后台控制, AutoCAD 作为前台图形显示, 开发了一种

(上接第 3164 页)

测试结果表明, 基于 D-VRMagic 设计的五子棋游戏具有 CPU 占用率低、内存开销小、网络通讯量少、操作延迟小等明显优点。

4 结论

本文在 VRMagic 的基础上, 结合高层体系结构(HLA), 设计并实现了一个通用分布式虚拟现实软件开发平台 D-VRMagic。文中重点讨论了多管理者多客户端网络架构、统一对象模型、管理者设计规范和通用客户端的设计过程。最后, 我们成功地开发了基于 D-VRMagic 的网络五子棋应用系统, 并以此验证了 D-VRMagic 的开发效率、可扩展性和通用性。必须指出的是, 作为一个原型系统, 目前的平台仍有待进一步完善。例如, 可通过借鉴商用开发平台的设计模式, 进一步增强平台的易用性。

参考文献:

- [1] Broll W. Distributed Virtual Reality for Everyone—a Framework for Networked VR on the Internet [C]// Proceedings of the IEEE VRAIS 1997, Las Alamitos, CA, USA: IEEE Computer Society Press, 1997, 121-128.
- [2] 吴宇, 康凤举, 魏佳宁, 等. 基于 HLA 的鱼雷武器作战性能仿真技术研究[J]. 系统仿真学报, 2006, 18(11): 3135-3142.

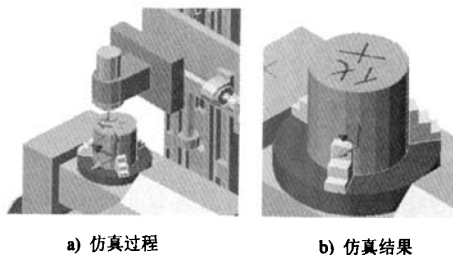


图 5 虚拟加工仿真

新型多功能车铣复合数控机床的加工仿真软件系统, 实现了 AutoCAD 造型与 CNC 仿真的有机结合, 完成高效 NC 代码驱动的实时加工过程仿真。通过该加工仿真软件对 NC 程序进行验证, 在计算机上利用三维图形技术对数控加工过程进行仿真, 可以快速高效地对 NC 程序进行正确性验证和评估, 降低生产成本, 提高工作效率。在仿真过程中画面连续, 切削流畅, 具有广泛的应用价值。

参考文献:

- [1] 李世国. AutoCAD 高级开发技术 ARX 编程及应用[M]. 北京: 机械工业出版社, 1999.
- [2] 杨胡坤, 徐步云, 梁宏保. 多坐标数控仿真技术研究[D]. 大庆石油学院, 2005.
- [3] 陈修龙, 赵永生. 虚拟轴机床数控加工过程的计算机模拟[J]. 系统仿真学报, 2004, 16(3): 541-545.
- [4] 梁宏宝, 等. 虚拟加工中 NC 代码转化技术研究[J]. 计算机仿真, 2004, 21(11): 219-222.
- [5] Valery Marinov. A Generic Virtual Machining Process [C]// Proc. The Third World Congress on Intelligent Manufacturing Processes & Systems, Cambridge, MA, USA, 2000: 203-208.
- [6] Ntasis E, Maniatis T A, Nikita K S. Real-time Collaborative Environment for Radiation Treatment Planning Virtual Simulation [J]. IEEE Transactions on Biomedical Engineering (S0018-9294), 2002, 49(2): 1444-1451.
- [7] Shaw C, Green M, Liang J D, Sun Y Q. Decoupled Simulation in Virtual Reality with the MR Toolkit [J]. ACM Transactions on Information Systems (S1046-8188), 1993, 11(3): 287-317.
- [8] Sense8 Corporation. WorldToolKit Reference Manual Release 9 [K]. Mill Valley, USA: Engineering Animation, Inc., 1999.
- [9] 赵沁平. DVENET 分布式虚拟环境[M]. 北京: 科学出版社, 2002.
- [10] 邓志东, 余士良, 张杨, 等. 通用虚拟现实软件开发平台的研究及其应用[J]. 系统仿真学报, 2006, 18(12): 3438-3443.
- [11] Defense Modeling and Simulation Organization. High Level Architecture (HLA) - Interface Specification, Version 1.3 [K]. Washington, D.C., USA: Defense Modeling and Simulation Office, 1998.
- [12] 周彦, 戴剑伟. HLA 仿真程序设计[M]. 北京: 电子工业出版社, 2002.
- [13] Department of Defense US. High Level Architecture Run-Time Infrastructure Programmer's Guide RTI 1.3, Version 6 [S]. 1999.
- [14] 陈静, 龚健雅, 朱欣焰, 等. 分布式虚拟地形场景仿真研究[J]. 系统仿真学报, 2007, 19(5): 1057-1081.
- [15] 刘雄, 康凤举, 王小宁, 等. 基于 HLA 编队协同作战仿真研究[J]. 系统仿真学报, 2007, 19(4): 784-786.
- [16] 凌云翔, 邱添珊, 史湘宁. 基于 HLA 的仿真网络系统设计问题[J]. 系统仿真学报, 2006, 18(1): 88-90.
- [17] 齐元胜, 王晓华, 郭俊忠. 印刷机分布式虚拟样机系统及实现策略研究[J]. 系统仿真学报, 2006, 18(51): 78-80.