

VRML 在结构模型分析及虚拟展示系统中的应用

中国人民解放军第四零九医院

周昕宇

中国人民解放军 37216 部队 82 分队

王 硕

▲ 本文详细阐述了 VRML 技术在结构模型分析与展示中的技术实现手段, 简要介绍了 VRML 规范的技术要求和使用方法, 并针对具体开发中出现的难点提出行之有效的解决方法和在设计中应注意的技术细节, 技术方案通用性良好, 便于扩展, 具有广阔的应用前景。

随着工业控制技术的飞速发展和科学技术分工的逐渐细化, 现代工业应用中大型机械及控制系统日益庞大和复杂。现行系统设计方法往往按照分布式原则先将整个系统细分成多个分系统并由相关的各职能部门和专业人员分别同时设计制造, 再将测试合格的各分系统拼装成总系统, 在设计过程中各分系统间的接口技术不规范, 协调性差, 极有可能出现匹配问题而直接影响总系统的最终集成精度; 各分系统在设计、分析、制造时常常因不了解与之相衔接的其他分系统的工艺情况和连接接口的技术定义而难以确定各自的设计方案。

另一方面, 分系统设计、测试时用现行实验手段进行分析研究的代价过高, 实验周期长, 实验中随机影响因素较多并常常与系统影响因素同时对测试系统施加影响, 造成实验结果难以判读, 无法迅速确定设计方案的有效性。尽管现已开发出多种微机模拟分析系统如 ANSYS 公司的有限元分析软件, 但价格高昂, 操作复杂, 在实验进行中用户不能介入实验过程和做出阶段性的效果判断, 现有实验模拟系统往往依赖微机的本地处理能力和系统资源, 不便于远程观察实验过程和及时分析实验结果; 数据存储结构封闭, 与其他数据系统的数据连接困难, 交互性差, 因此难以满足分布式集约化系统开发的技术要求。

而有些微观物理实验如分子级的物理结构分析和展示难以用廉价的实验手段表述, 这就要求提供一种高效而低成本的物理结构分析与展示手段。VRML (Virtual Reality Modeling Language, 虚拟现实建模语言) 技术是 SGI 公司提出并经国际标准化组织 ISO 批准的一种网络环境下描述虚拟三维模型和提供交互操作的跨平台的开放式解释性文本语言, 可以在低带宽异构网络环境中实时进行三维信息的远程传输和远程控制指令的接收, 能与其他数据系统顺利连接, 提供了良好的人机界面和完善的对象结构模型, 较好的满足了结构模型系统远程演示与简单分析的技术要求。本文就有关技术实现手段及相关的设计中应注意事项进行讨论, 仅供参考。

一、VRML 空间与外界交换信息的机制

VRML 空间通过两个特殊的节点 (script 节点和 inLine 节

点) 来与外界交换信息和在空间内部对信息作出处理, 组合使用这两个节点将极大的提高三维虚拟空间的信息处理能力。

1. script 节点

script 节点定义了用户编制脚本的指令域, 接收外部数据的输入域和向外界发送数据输出域。Script 节点的语法为:

```
Script { url []  
    field eventIn fieldtype fieldname  
    field eventOut fieldtype fieldname  
    mustEvaluate # field SFBool  
    directOutput # field SFBool}
```

在 url 域中用户可以编制对虚拟模型本地操作的脚本, 在这里允许用户使用 java/javascript 语言进行编程, 但在脚本最前面向 VRML 解释器声明所使用的语言。

EventIn 域定义了触发器的输入接口和初始状态, 一般以 "set_" 开头, 通过它节点可以感受到外界和其他模型的空间操作并将信息送入封装在 url 域中的脚本中进行处理。

EventOut 域定义了触发器的输出接口和初始状态, 一般以 "_changed" 结尾, 节点可以通过它将本地脚本的运行结果和操作信息记录在自定义的变量中并向外界或其他空间模型发送。

在编制用户自定义脚本和设定虚拟模型的传感事件时, script 节点的输入、输出域和其他感受器节点或可见三维模型节点的输入、输出域的数据类型必须严格一致, script 节点还包含 mustEvaluate 域和 directOutput 域, 分别设置 VRML 空间中三维模型在接收到外界输入的新值时是否立即调用程序脚本和是否允许脚本直接控制其他节点的空间状态。

通过与其他传感器进行接口连接, Script 节点建立起了 VRML 空间与外界信息交互的有效通道并丰富了虚拟模型对用户操作的处理能力, 使虚拟空间具有更好的自主判断能力和更多的感知能力。

2. inLine 节点

inLine 节点允许用户往一个 VRML 空间中导入原本存在于另一个 VRML 空间中的虚拟模型, 同时保留该模型自身所

携带的全部传感器和内插器。在导入时它会自动将引入空间的三维坐标匹配为当前空间的世界坐标,从而保证了模型在新的虚拟空间中的正常显示比例和与其他模型间的三维位置关系。其语法为:

```
inLine { url [] #Mfstring
    Bboxcenter    #Sfvec3f
    Bboxsize      #Sfvec3f }
```

其中url域标明了将要导入的VRML文件在网络环境中的逻辑存储地址,对其进行修改时可使用隐含eventIn域“set_url”。而bboxcenter域和bboxsize域标明了导入文件所原有的三维空间在导入到新空间后的中心位置和大小。

下面以向已有空间内导入新的VRML模型为例演示VRML空间对用户操作的反应能力:

首先在已有空间中设定一个script节点用来判断用户操作的状态并发出是否导入新模型命令:一个inLine节点负责在网络中提取新模型,同时在VRML空间中安装一个touchSensor节点感知用户的点击触发。由于touchSensor节点只能感应到指点设备当前的触发状态并由此改变自身的boolean状态值却不能自动记录触发状态,因此在script节点中定义了用户脚本来对触发事件的状态进行记录和判断脚本,同时根据判断结果向inLine节点指示是否导入新的虚拟模型。源代码片段如下:

```
# 创建一个接触传感器
DEF touch touchSensor { }
## 创建 script 节点
DEF judgement script {
    url"javascript: ## 声明所使用脚本语言的种类
    function set_boolean(boolean) {
        if (boolean==false){return}
        if (value==true){record="url";## 声明一个中间变量用来记录欲导入的VRML 空间文件在网络环境中的逻辑存储地址
        Value=false }
        else {record=""
        Value=true }
        Value_changed=record
    }"
    MustEvaluate true ## 允许 VRML 解释器立即调用脚本
    DirectOutput true ## 允许脚本直接修改其他节点的三维坐标和外形状态
    Field SfBool value False
```

```
eventIn SfBool set_boolean
eventout Mfstring value_changed
}
## 创建 inLine 节点
DEF newWorld inLine {
    url [] # Mfstring ## 指向欲导入的VRML 空间文件在网络环境中的存储地址
    bboxCenter # Sfvec3f
    bboxSize # Sfvec3f
}
## 对节点进行接口连接
rout touch.isActive To judgement.set_boolean
rout judgement.value_changed To newWorld.set_url
```

二、VRML 空间与数据库的交互

在有限元分析中经常需要修改模型的物理形状和机械参数,因此用VRML空间分析虚拟模型的特征参数时要求模型可以随时接收外界的数据和修改命令并实时显示更改后的状态,由此就要求VRML空间必须与数据库建立良好的数据传输通道。

1. VRML 虚拟模型的定义要求

尽管允许虚拟空间拥有自主判断和响应能力,但基于网络环境的VRML规范要求空间解释器逐行解释源代码,因此客户端系统副本运行效率低,反应速度慢,不适合进行大规模数据采集和运算而更适合用来显示运算结果。因此在编制VRML文件时应尽量少封装对整个虚拟空间进行坐标操作的控制类节点,模型自带的传感器以规范定义的标准感知器为主,所有必须的判断事件应尽量集中在script接点中,script节点应定义好虚拟模型与外界进行数据传递的接口标准并与虚拟空间中的其他感知器建立有效连接。

在描述物体外形时VRML规范只定义三角形面片(face)的三个顶点并进行矢量连接,不支持点的张力效果,造成模型外观粗糙,棱角明显。VRML技术规范不支持某些纹理贴图(如镜面反射、凹凸贴图等)和特殊环境效果(如体光、体雾等),对模型外形的描述手段较少。因此在对模型细节进行分析时应减少VRML空间中表述模型外形三角形面片的数量和与结构分析无关的对模型外形的细节描述以及对虚拟空间的环境设定而尽量用贴图来进行简单描述,这样即可以保证对模型外观的准确表述又可以提高显示效果和响应速度。

VRML规范是基于文本的,所以在理论上讲任何一种文本编辑器都可以编写VRML文件。而kenitex公司开发的3D

Studio MAX 软件是现时较流行的三维建模软件, 它可以通过内嵌的VRML转换器将3DS MAX格式的模型转换为VRML格式 (VRML转换器可以在该公司的网站上免费下载, 网址为 www.kenitex.com)。而在3D Studio MAX 2.0及更高版本中更内置了大量VRML传感器, 可以快捷的建立三维模型并规划模型间的相互关系。

为保证与常用开发软件的衔接, 可以先在AutoCAD等专业设计平台上建立三维模型再将其导入3D Studio MAX软件中, 此时应注意, 由于AutoCAD是用矢量公式表述模型的外观, 而3D Studio MAX则是用三角形面片表述模型外观 (这些面片通过公用顶点相互连接并具有张力特性), 因此直接导入会导致大量增加模型的面片数并导致最终的VRML文件过于庞大, 因此在导入模型后应分析其面片分布并用简单的三维基本造型进行局部优化。对具有不规则外形的模型尽量用放样方式和模型拼接方式来表述而避免用布尔 (boolean) 运算制作, 否则将增加大量不必要的面、线。

对模型优化后向其嵌入各自的传感器并进行接口连接。此时应注意利用组 (group) 集合三维模型并设置详细的动作过程。之后将3D Studio MAX模型导出为满足VRML2.0规范的VRML模型。在文本编辑器或Microsoft公司开发的Visual Studio 6.0软件中调试javascript脚本, 之后将其封装再VRML模型的适当位置并与其他VRML模型中的感受器进行接口连接, 从而最终确定了VRML文件的全部内容。

2. 数据库的结构要求

数据库一般用来记录模型的全部技术参数值和空间中的位置状态, 它能够方便的查询出各项数据, 同时应具有实时接收数据并更新数据库本体的能力。为适应网络环境下的分布式数据存储和结构化数据查询以及远程集约化管理的要求, 数据库选用SQL体系为宜。

数据库可根据实际情况分解成若干个在逻辑上相对独立而在物理位置上相互分离的子数据库, 各子数据库本体中的数据结构以分层的树状结构为基础, 同层数据间以事先确定的变量标志相连接。虚拟模型的所有参数作为数据页中的列值 (或表单中的字段) 记录在数据库本体中, 每个列都建立统一的索引 (这个索引在整个数据库中必须是唯一的), 列与列之间通过键值连接, 这些键值必须是在整个数据库中唯一并严格一一映射的, 从而实现跨数据页和数据库的逻辑联系。

在对多媒体信息存储时应避免将其直接存储在数据库本体中, 否则不仅数据格式转换效率低, 而且数据库自身也消耗了大量的宝贵资源。应尽量将多媒体信息独立于数据库存储在网络中不同的物理位置并进行相对集约的逻辑

管理, 在数据库中只建立数据指针来记录数据的在网络环境中逻辑存储位置并将指针封装为数据页中的列。当用户需要访问这些信息时, 数据库将数据指针反馈给用户终端, 在用户终端运行的系统副本按指针内容从网络中调用所需的信息。数据库只维护数据指针而不介入在用户终端运行的系统副本对多媒体信息的调用事务。

3. 中间数据通道的建立

由于VRML空间不具有直接与数据库进行信息交互的能力, 因此在VRML空间与数据库之间必须建立高效的中间数据通道。首先, 数据库将数据 (包括数据指针) 传输到中间连接通道中, 而中间连接通道将数据记录在内存变量中, VRML空间再将内存变量里的数据读入并由此决定显示效果。同时, 中间数据通道可以将用户从客户端界面输入的更新数据存储到内存变量中送到VRML空间中实时显示并将数据库中相应的数据更新。

基于网络环境的中间数据通道选用ASP (ActiveX Server Page) 编制较合适。ASP本质上是服务端运行的ActiveX技术, 它也是基于文本的解释性语言, 可以方便的与数据库连接并进行数据交互。同时, 它内嵌的session对象和response对象可以高效的定义并记录用户将要使用的对象并将获得的数据存储到一个全局变量中。ASP运行在服务器上, 在网络中只传递所需的数据和运行结果, 因此网络介质中传输的数据包少, 服务器和用户终端的资源开销都较小, 可以保证数据响应的高效性和数据传输的准确性。

4. 中间数据通道的接口连接

数据交互时, ASP将数据库中传输过来的数据存储在变量中, 按事先设定将该变量传输到VRML空间中script节点的输入接口, script节点将变量携带的数据提取出来, 经运算处理和判断将处理结果从输出接口传递到VRML空间中其他节点的接口域, 虚拟模型中封装的感受器从各自的感受接口接收数据并作出相应的属性变化。

ASP可以接收用户终端发出的修改命令, 提取用户从客户端界面输入的数据和指令, 经确定操作的有效性后将变化数值传递进VRML空间, 此时这些操作都是由在本地运行的系统副本来实现的, 用户终端与数据库服务器的数据通道是关闭的。ASP也可以将用户提供的本地数据经必要的安全认证写入数据库, 此时在网络中传输的只是数据和经过有效性确认的操作指令, 用户终端上运行的系统副本与数据库服务器经Web服务器隔离, 不能直接介入数据库的更新事务从而保证了数据库的安全有效。如果在用户终端运行的系统副本要同时提供与用户交互的HTML界面和显示三维虚拟空间的VRML界面并要求系统优

先满足数据进行远程传输时的安全可靠和对远程控制指令的及时响应时, 可以将系统界面设计成多帧模式。由于 VRML 解释器不能与 HTML 解释器同时有效而写屏效率又远低于 HTML 解释器, 因此应将虚拟空间界面帧中的 VRML 解释器内嵌在 HTML 解释器中, 具体实现方法为将 VRML 语句用 `writeln` 语法封装并将其作为标准 javascript 语句写入脚本, 这样在终端运行的系统副本先利用 HTML 解释器进行高效的脚本解释, 再根据 `writeln` 语句中声明的 MIME (Multipurpose Internet Mail Extension, 多用途 Internet 邮件扩展) 类型调用 VRML 解释器, 这样在提取 VRML 语句的同时就可以将数据库中的数据存储在变量中直接传递到 VRML 解释器中从而与 VRML 语句一起解释、写屏。

三、小结

VRML 技术可以低成本的模拟各种与物体外形和空间状态有关的实验并实现了对虚拟试验的过程中介入, 允许用户对虚拟模型的自由角度观察。同时还提供了有效的与其

他外部数据源进行交互的接口, 实现了对结构模型和机械系统运行的远程实时观察和低代价的集约化开发测试。尽管现行技术规范的运行效率和显示效果较差, 但随着网络通信技术的高速发展, 技术规范的扩充和标准化, 以及三维图形加速卡写屏效率的进一步提高, VRML 空间将更加丰富多彩, 显示效果将更真实, 与客户的交互能力将更强大。VRML 技术必将越来越多的应用于宇航研究、军事模拟、机械设计分析以及医药化学物质结构分析等各种虚拟环境的演示和技术探讨中。

参考文献

- [1] 《VRML 资源手册》(美) Andrea L. AMES 等著电子工业出版社 1998 年版
- [2] 《Visual Interdev 6.0 开发指南》(中) 任伟等著北京航空航天大学出版社 1999 年版
- [3] 《MicroSoft SQL 6.5 管理员手册》(美) MicroSoft 公司著科学出版社 & 龙门书局 1997 年版