

火箭炮虚拟漫游系统的设计与实现

杨清文, 郑 义, 王晓敏, 李慧翔

(解放军炮兵学院一系, 合肥 230031)

摘 要: 针对火箭炮兵器结构教学存在的实际困难, 提出运用虚拟现实技术开发一套火箭炮虚拟漫游系统。系统采用 B/S 结构, 运用 3DSMAX 和 VRML 联合建模的方法创建虚拟场景, 并对其优化, 通过加入摄像机视点和悬浮菜单实现对场景的导航, 运用 ASP+Access2000 搭建网络平台, 构建一个生动直观、丰富多彩的虚拟现实的漫游学习系统。

关键词: 火箭炮; 虚拟现实; 虚拟现实建模语言; 漫游

Design and Realization of Rocket Launcher Virtual Walkthrough System

YANG Qing-wen, ZHENG Yi, WANG Xiao-min, LI Hui-xiang

(First Department, Artillery Academy of PLA, Hefei 230031)

【Abstract】 Aiming at the structure teaching problem of the rocket launcher structures, the rocket launcher virtual reality walkthrough system using virtual reality technique is put forward. The system adopts B/S structure, and the virtual scene is applied and optimized by combining 3DSMAX and Virtual Reality Modeling Language(VRML) to modeling. Navigation of the scene is realized through joining camcorder viewpoint and suspension menu. Network platform is built with ASP+Access2000. A vivid virtual reality walkthrough learning environment is constructed.

【Key words】 rocket launcher; virtual reality; Virtual Reality Modeling Language(VRML); walkthrough

火箭炮系统零部件多、结构复杂、原理抽象, 给兵器结构原理的教学带来了极大的困难, 以往教员利用挂图、模型进行讲解的方法已不能满足实际的教学要求。随着虚拟现实技术的发展, 给教学训练对象提供一个实时交互和有较强沉浸感的学习漫游系统成为了可能。为此, 本文采用虚拟现实、网络、数据库等技术和手段, 建立一套适应新装备教学的信息化虚拟漫游教学系统。

1 系统设计

1.1 系统结构

如图 1 所示, 整个系统流程划分为 3 层, 分别为数据库管理层、应用层及交互层。

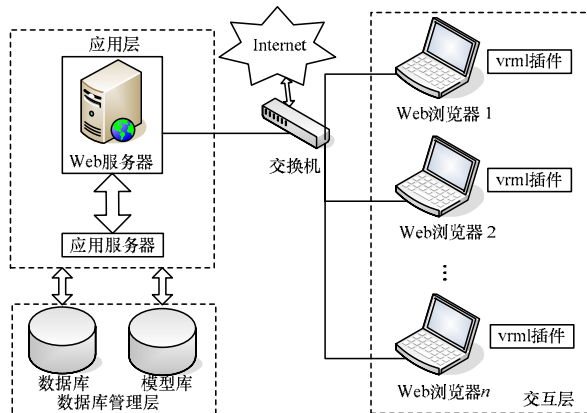


图 1 系统体系结构模型

系统采用 B/S 模式。系统对用户端软件进行了简化, 只需装一个 VRML 浏览器插件作为应用的运行平台, 所有的开发、维护、升级工作都集中在服务器端。客户端零维护, 并

且系统的扩展非常容易。这样简化了客户端载荷, 减轻了系统维护与升级的成本和工作量。学员只要有一台能与校园网相连的计算机便可在任何地方进行操作^[1]。

1.2 系统组成及功能

系统主要由武器系统介绍模块、单体学习模块、智能搜索模块、在线提问答疑模块及后台管理模块组成。可以实现武器漫游、单体三维演示、重要部件分解结合的训练、数据管理、用户信息管理等功能。系统组成框图如图 2 所示。

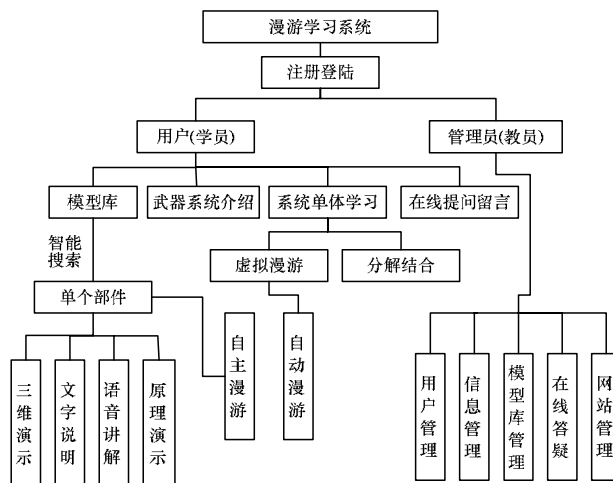


图 2 系统组成框图

作者简介: 杨清文(1966 -), 男, 教授、博士, 主研方向: 武器系统动力学仿真; 郑 义, 硕士研究生; 王晓敏, 讲师; 李慧翔, 硕士研究生

收稿日期: 2009-10-17 **E-mail:** zhy2836@163.com

1.3 系统开发及运行环境

该系统可运行于校园局域网或 Internet 环境，以 Access2000 作为数据库开发平台并与微软操作系统、Web 服务器 IIS5.0 实现无缝集成；采用 ASP 结合 VRML 和 JavaScript 作为开发语言；采用 Dreamweaver8, Fireworks8 进行页面设计，并结合 3DSMAX 等工具进行辅助设计。

2 系统主要功能实现

2.1 虚拟场景的建立

为了使用户能够真正沉浸于一个由计算机生成的虚拟环境中，必须使生成的环境足够逼真和自然。虚拟场景的构建主要完成的是场景背景的设置、三维模型的构建和场景的优化等工作。其中，三维模型的构建是漫游系统实现的基础。

2.1.1 场景背景的设置

设置背景是通过设定 Background 节点各域的参数来实现的，可分为 2 种设定方式，一种是通过颜色插值模拟大地和天空；一种是构造背景的全景图。本系统采用后者。把全景图看成是一个无穷大的立方体，由 6 幅图片组成，分别位于立方体的 6 个面^[2]。因此可以实现纹理映射的无缝粘贴，确保环境模型纹理图像的连续性，创建了一个较为真实的空间背景。

2.1.2 三维模型的构建

三维模型采用 VRML 和 3DSMAX 联合建模的方法。创建的步骤如下：

(1)数据采集与加工。为了使场景真实，需要对实装采集一些数据，包括实装的尺寸、重要部件的设计图纸及拍摄各角度的数码照片。对采集到的图像用 Fireworks8 等工具软件进行加工处理，使其更适合贴图。

(2)用 3DSMAX 建造基本场景。整个火箭炮系统的建模，工作量大而且复杂，用 VRML 编写代码的方式去完成建模是不适合的。本文选用 3DSMAX 来搭建基本场景，其优势能够快速高效地构造复杂的三维模型，并且很容易添加材质、光效和视点，同时还具有很重要的一个功能就是模型文件能被导出为 .wrl 格式文件。图 3 为火箭炮炮车的三维模型。

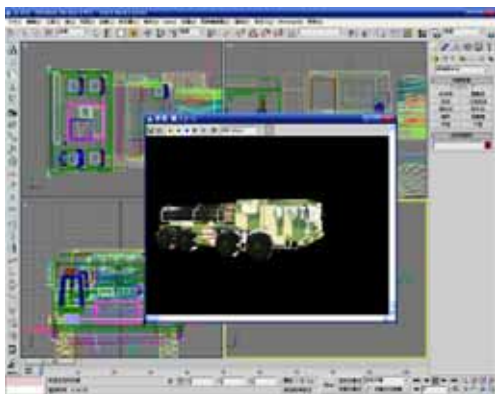


图 3 炮车三维模型

(3)用 VRMLPAD 完善虚拟场景。从 3DSMAX 导出的文件，由于交互行为不足，因此还应该打开生成的文件，对其代码做进一步的编写、完善。VRMLPAD 是一个很出色的虚拟现实文件专用编辑器，不仅有利于提高代码编写效率，而且还提供程序调试功能，是完成复杂虚拟场景的一个非常有利的工具。

2.1.3 场景的优化

.wrl 格式的文件由于 VRML 语言的开放性和编码的规范

性，导致其场景文件的数据量相对来说比较大，不利于文件的网络传输，因此完成 VRML 场景的创作后对文件进行优化压缩是很有必要的。采用以下几种优化方法：(1)通用设定可视距离，减少浏览器一次渲染对象的数量。VRML 的 NavigationInfo 节点的 visibilityLimit 域提供了对对象场景的可视化距离的设定。(2)最大可能地使用重用机制。对相同的部分利用 DEF 与 USE 可以极大地简化文件，减轻浏览器的负担。(3)利用 Inline 节点减少文件的体积，提高代码的重用率。(4)利用 Switch, LOD 节点实现场景切换技术及同一对象模型的自动转换。为场景中某一对象建立多个不同的模型，当视点离对象较远时采用简单模型，较近时采用细化模型^[3]。

通过以上方法，场景文件被大大压缩，由原来的 7.5 MB 压缩到了 1.43 MB。图 4 为驾驶室场景。



图 4 驾驶室场景图

2.2 漫游功能的实现

2.2.1 用户自主漫游

采用创建摄像机视图的方法来实现自动漫游。一方面，使用建立的摄像机可以展示最佳场景或者展示场景中重点部件的特写。另一方面，选择摄像机视图，就可以很好地控制希望用户看到的场景。Viewpoint 节点在 VRML 文件中直接控制场景中视点的方向很困难，因为在 Viewpoint 节点中必须给出一个空间的旋转轴和旋转角度，而在 3Dmax 中可以直接对一个摄像机进行所需要的调节，直到最后得到满意的空间视点，实现真正的所见即所得。在场景中所建立的不同摄像机移动路径可以在悬浮菜单上列出^[4]。用户可以通过选择悬浮菜单上的按钮在场景中导航。

悬浮菜单也可称为导航菜单，即在场景中显著位置的一组按钮，用来选择场景中不同的视点。悬浮菜单的创建主要是利用位置触发器 ProximitySensor 节点的属性。按钮通过 box 节点在场景中建立。ProximitySensor 节点指定一个可感知的区域，当观察者的位置和方向改变时，Position_changed 和 Orientation 域(eventOut 出事件)输出新的位置和方向，将此值传给 box 节点，就可以实时改变按钮的位置，达到按钮始终处于场景固定位置的效果。在按钮造型中加入 anchor 节点，使其链接对应的视点。代码如下：

```
DEF Where ProximitySensor {size 60000 60000 60000}
DEF Main Transform {
  children Transform {
    children Anchor {
      ...
      url "#Camera04" }}}
ROUTE Where.position_changed TO Main.set_translation
ROUTE Where.orientation_changed TO Main.set_rotation
```

2.2.2 自动漫游的实现

自动漫游功能使得用户不依靠导航按钮也可完整地游历武器系统。具体实现方法是：建立触发检测节点 Tour，当它被触发时，激活名为 Camera-Timer.startTime 的域，从而开始

动画的计时, 由此每次 Camera-Timer.fraction_changed(一个浮点值, 即每隔一个周期由 0 到 1 产生一个单调上升的值直到 1 为止)都会分别激活 Camera-POS.set_fraction 及 Camera-ROT.set_fraction。Camera-POS 是一个位置插补器, 作用是当收到输入接口的 set_fraction 时, 在 key 域中查找其值, 并通过 value_changed 输出接口将该值输出。其输出数据是一个 SFVec3f 类型数据, 由 3 个浮点数构成, 表示一个位移值。Camera-ROT 与 Camera-POS 相似, 是一个方向插补器, 它的输出是一个 SFRotation 类型数据, 由 4 个 0~1 之间的浮点数组成, 主要表示一个方向。当 Camera-POS 和 Camera-ROT 分别将值传到 Camera 视点的 position(位移)及 orientation(转向)时, Camera 视点便会产生移动的效果了。

2.3 碰撞检测的实现

本系统利用化身来进行碰撞检查。浏览者在漫游一个场景时, 浏览器都会创建一个浏览者的化身, 该化身是不可见的, 浏览器所展现的景物是以化身的视角来观察的。在设计中, 使用 NavigationInfo 节点的 avatarSize 域来设置场景中浏览者的化身尺寸。代码如下:

```
NavigationInfo {
  avatarSize [0.1, 1, 0.5]
  ...
}
```

avatarSize 定义了一个不可见的圆柱体, 如图 5 所示。它有 3 个参数, 第 1 个参数用于指定化身与其他几何体发生碰撞的最小距离, 相当于圆柱体的半径; 第 2 个参数指定化身头顶距地面的高度; 第 3 个参数用于指定浏览器能够跨越的最大高度^[5]。在本系统中, 当浏览者与物体的距离小于 avatarSize[0]即 0.1 m 时, 将触发碰撞事件。如果障碍物的高度大于 avatarSize[2]即 0.5 m, 则浏览者将无法跨越, 利用这一点可以限制浏览者活动的范围。

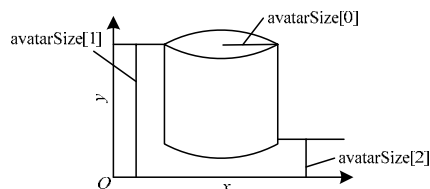


图 5 avatarSize 图示

(上接第 262 页)

种, 利用 FPGA 来实现防碰撞算法, 是很可取的一种思路。在后续研究中, 将对系统运行的稳定性进行深入探讨, 并考虑防碰撞模块与阅读器其他模块之间的相互接口以及相互干扰等问题。

参考文献

[1] Klaus F Z. 射频识别技术[M]. 3 版. 吴晓峰, 陈大才, 译. 北京: 电子工业出版社, 2006.
[2] ISO/IEC. ISO/IEC 14443-3 Initialization and Anti-collision[S]. 1999.

2.4 漫游场景数据库的开发

三维建模后得到的场景模型是以场景数据库的方式进行管理和操作的。本虚拟漫游系统面向学校的数据中心服务器, 所以选用小型的数据库。在与数据库的连接中选用的后端数据库管理系统是微软的 Access2000。

使用 ASP 的 RecordSets 通过 ADO 访问数据库的运行机制如图 6 所示。

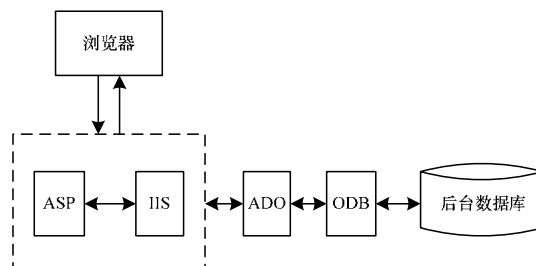


图 6 ASP 运行机制

ADO 使用内置对象作为数据的主要接口, 使用 JavaScript 语言来控制所有支持 ODBC 数据库的访问, 以及查询结果的输出显示。

3 结束语

本文根据火箭炮结构教学的要求, 运用虚拟现实技术、网络技术及数据库技术设计实现了基于虚拟现实的火箭炮漫游系统。该系统具有虚拟漫游、人机交互、信息管理等功能, 对改变传统兵器结构教学和训练有积极的意义。

参考文献

[1] 卞步喜. 虚拟现实技术在工程力学网络教学中的运用[D]. 合肥: 合肥工业大学, 2003.
[2] 阳化冰, 刘忠丽, 刘忠轩, 等. 虚拟现实构造语言 VRML[M]. 北京: 北京航空航天大学出版社, 2000.
[3] 宋慧玲. 基于 VRML 的地质虚拟博物馆漫游系统的实现[J]. 计算机工程, 2006, 32(17): 239-241, 248.
[4] 丁 妹, 胡志秋. 虚拟建筑模型场景漫游系统[J]. 计算机技术与应用, 2005, (1): 46-48.
[5] 刘 爽. 基于 VRML 的虚拟现实场景漫游技术的研究与实现[D]. 长春: 吉林大学, 2007.

编辑 任吉慧

[3] 谢振华. 标签防冲撞算法设计[J]. 计算机工程, 2008, 34(6): 90-92.
[4] Ning Huansheng, Cong Yu, Zhang Yan, et al. Performance Evaluation of RFID Anti-collision Algorithm with FPGA Implementation[C]//Proc. of AINAW'07. Niagara Falls, Canada: [s. n.], 2007.
[5] 李兴鹤, 胡咏梅, 王华莲. 基于动态二进制的二叉树搜索结构 RFID 反碰撞算法[J]. 山东科学, 2006, 19(2): 51-54.

编辑 陆燕菲