

基于 VRML 与 Java3D 的交互式虚拟物流系统

刘海霞, 李仁旺, 杨 贵, 张鹏举

(浙江理工大学工业工程系, 杭州 310018)

摘要:为实现虚拟物流系统的实时交互,研究 Java3D 与 VRML 的结合方法。通过 3D 软件建立物流设备图,由 VrmIpad 导入文件并进行优化,再由 Loader 接口将 VRML 造型文件导入到 Java3D 场景中。借助数据库和 Java 语言的事件处理器模型以及 Java3D 的 Behavior 类,实现人机交互、虚拟漫游以及虚拟场景动画与库存的双向驱动。通过虚拟物流系统实例验证了该方案的有效性。

关键词: VRML 语言; 建模; 人机交互; 物流系统

Interactive Virtual Logistics System Based on VRML and Java3D

LIU Hai-xia, LI Ren-wang, YANG Gui, ZHANG Peng-ju

(Dept. of Industrial Engineering, Zhejiang Sci-Tech University, Hangzhou 310018)

【Abstract】To develop the interaction of virtual logistics system, the technology based on VRML and Java3D is discussed. The virtual logistics equipments are built by 3D software. These equipments are loaded to VrmIpad and optimized. The files of the equipments are leaded into Java3D scene via the loader interface. The interactive virtual assembly and the bidirectional drive between the virtual scene animation and the stocks are implemented by using the Java event-handling model, combined with the database technology. An example about virtual logistics system demonstrates that the scheme is effective.

【Key words】 Virtual Reality Modeling Language(VRML); modeling; man-machine interaction; logistics system

1 概述

虚拟现实系统的开发涉及虚拟场景构建、可视化及声效处理、人机交互等技术。以往的虚拟现实建模一般采用 OpenGL, VRML, Driect3D, Java3D 等语言。但用 OpenGL 与 Driect3D 编写复杂的三维图像非常繁琐;虚拟现实建模语言(Virtual Reality Modeling Language, VRML)在网络上得到广泛应用,但其本身的交互控制不适合构建大型交互式系统,且不支持 Java 语言 1.4 以上的版本。

Java3D是Java语言在三维图形领域的扩展,也是Java的一个API库,是一套面向对象的开发工具^[1-2]。但要实现复杂的3D图形建模,代码编写的工作量太大。因此,本文中的虚拟物流系统采用VRML与Java3D相结合的方法来构建虚拟场景,并通过Java语言实现交互操作。

2 系统整体设计

现代物流与 Internet 网络技术的结合与应用,使物流实体网络与物流虚拟网络的结合日趋完善。今后将出现许多不具有运输工具及仓库,而以一体化整合为主体的提供物流服务的虚拟物流组织或企业,这种新的物流组织已成为当前物流发展的趋势,越来越受到物流界的重视。

该系统是基于 VRML 和 Java3D 技术来实现虚拟物流企业仓储管理功能的。通过 Java 语言来控制三维场景的交互,由数据库进出货物流来驱动虚拟场景的动画操作,实现用户与虚拟场景之间,以及用户与用户之间的交互及共享。

系统主要是针对虚拟物流业的仓储方面来设计的,包括库存、三维动画及其交互漫游操作,采用 MVC(Model View Controller)设计思想,即把一个应用的输入、处理、输出流程按照 Model, View, Controller 的方式进行分离,这样,一个应用被分成 3 个层:(1)模型层;(2)视图层;(3)控制层。

视图(view)代表用户交互界面。模型(model)是业务流程/状态的处理,以及业务规则的制定。业务流程的处理过程对其他层来说是黑箱操作,模型接收视图请求的数据,并返回最终的处理结果。

业务模型的设计是 MVC 的核心。控制(controller)可以理解为从用户处接收请求,将模型与视图匹配在一起,共同完成用户请求。

实现模型与 MVC 设计模式分别如图 1、图 2 所示。

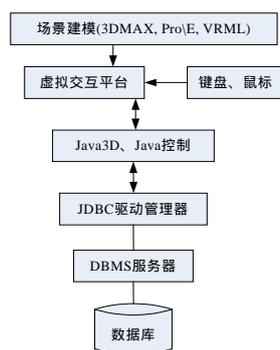


图1 系统实现模型

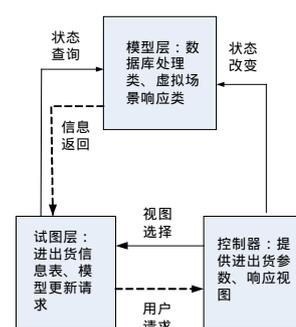


图2 系统 MVC 设计模式

3 实体建模

3.1 实体建模工具

VRML把二维、三维、动画、多媒体和人机交互多种技

基金项目:国家自然科学基金资助项目(50675208);浙江省自然科学基金资助项目(Y605187)

作者简介:刘海霞(1981 -),女,硕士研究生,主研方向:先进制造技术;李仁旺,教授、博士后;杨 贵、张鹏举,硕士研究生

收稿日期:2007-12-13 **E-mail:** wmlhx8866@yahoo.com.cn

术融合为一体,能满足用户多方面的要求。VRML对三维场景有很强的描述能力,它把现实世界抽象为节点。节点是VRML文件最基本的组成要素,节点由不同的域组成,域值即为场景特征^[3]。场景模型主要包括模型的几何形体、外观特征和位置。

为了有效地实现VRML文件在网络上的浏览,以及能够构建大型系统的交互性,笔者采用VRML与Java3D相结合的方法,完成虚拟现实的场景构建、交互操作、虚拟漫游。Java3D通过Loader接口导入VRML文件,加入到场景图中,利用丰富的Java3D API实现对三维实体的操作。

3.2 Java3D 场景的结构

Java3D是Java语言在三维图形领域的扩展,与Java一样,Java3D也是纯粹的面向对象结构。Java3D场景结构图SceneGrpStructure如图3所示,它是一个有向无环图。

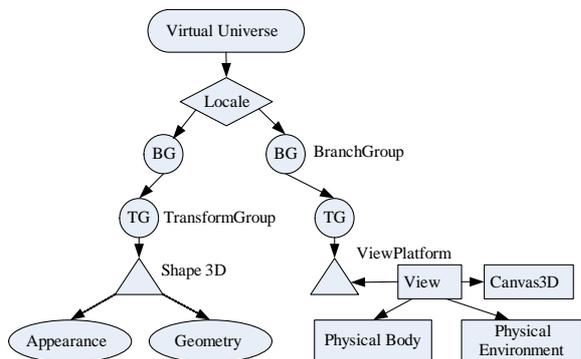


图3 Java3D 场景结构

从图3中可以看出:

(1)在Java3D场景图中,根部节点是VirtualUniverse,每个场景图只能有一个VirtualUniverse。

(2)VirtualUniverse的下面是Locale节点,每个程序可以有一个或多个Locale,但同时只能有一个Locale处于显示状态。就好像一个三维世界非常大,有很多景点,但同一时间只能在一个景点进行观察。

(3)每个Locale下面拥有一个或多个BranchGroup节点。在它下面建立一个基准坐标系TransformGroup,它用来摆放所需的形体,摆放时要指出形体的外观(Appearance)和几何信息(Geometry)。

(4)安放好三维形体后,在ViewPlatform处设定人的观察位置及观测参数。

3.3 虚拟场景的构建

虽然VRML提供了完善的建模节点,但要建造复杂的三维实体,其工作量还是巨大的。成熟的3D软件,如Pro/E,UG,3DMAX,I-DEAS等都支持*.wrl格式,用3D软件来建立模型,再导出VRML格式的*.wrl文件,在VRML编译环境下进行优化,这样,既保证了模型的逼真性,又提高了系统的开发效率,再通过Java3D导入到Java编程,构建整个虚拟场景。

整个系统的模型可分为两大部分:(1)实验设备;(2)放置实验设置的场地。物流设备大部分为机械设备,因此,选用Pro/E Wild Fire作为建立设置模型工具,可得到几何形体较为精确的模型。实验场所即室内环境,可由3DMAX完成。模型建立后,以*.wrl格式导出,在VrmlPad2.1编辑环境下,把两个文件整合为完整的虚拟场景模型。建模的流程如图4所示。

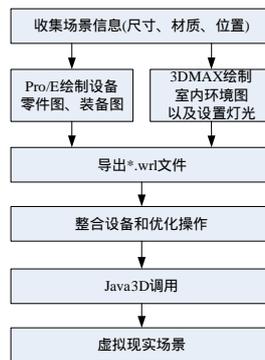


图4 建模流程

3.3.1 设备模型的建立

对各个设置进行拆分,绘制零件图,基本步骤如下:

- (1)建立或选取基准特征:作为模型空间定位的基准,如基准面、基准线、基准坐标系。
- (2)建立基础实体特征:拉伸,旋转,扫描,混合。
- (3)建立工程特征:孔,倒圆角,倒角,拔模,筋。
- (4)特征修改与操作:特征阵列,特征复制,特征。
- (5)添加材质和渲染处理。

由于要得到的是设备整体图形,因此在拆分时,不必按机械设计中的测绘要求,绘出每个零件的三维图形。在绘制过程中,可删减模型的细节,如螺栓连接。在不影响外观视图的情况下,可把几个零件合并为一个,以减少装配时的零件个数。

3.3.2 文件优化

Pro/E绘制设备零件图,3DMAX绘制室内环境图,然后导出*.wrl文件,为便于优化,为每类文件建立各自的文件夹。用VrmlPad导入VRML文件,主要通过两种方式进行优化。

(1)基于语言描述的优化

由于VRML文件是由其他3D软件转换过来的,因此语言描述及结构有很多重复,许多定义的标识符不会用到。对于这些有定义但又没有用到的语句,可以删除,以减轻文件重量。

VRMLPad提供这样的优化方式:在VRMLPad中,点击“工具”中“Commands”下的“Optimize Code”等一系列优化命令,可对文件进行优化,也能使用工具栏下“未使用标示符”,选取“所有未用标示符”,显示所有定义了又没有使用的语句,进行删除。用这种方法能够对设备装配图场景进行优化。

(2)基于语言结构的优化

对于文件中重复使用的相同属性节点,可使用实例(instance)重用,对已用DEF字定义的节点可以使用USE字进行重用。

3.3.3 Java3D调用优化后的*.wrl文件

将从VRML文件中读出的零件造型放置在TransformGroup对象中,以便变换坐标,然后,用addChild方法将TransformGroup对象添加到Java3D的三维虚拟世界中,该流程的代码如下:

```
Canvas3D c=new Canvas3D(config);
//创建画布 Canvas3D,从而构造出一个图形环境
{
SimpleUniverse Universe=new SimpleUniverse(c);
//创建虚拟
Universe.getViewingPlatform().setNominalViewingTransform();
```

```

//获得视点对象
BranchGroup objRoot=new BranchGroup();
//创建用来包含对象的数据结构
Universe.addBranchGraph(objRoot);
//将 objRoot 添加到虚拟世界
VrmlLoader l=new VrmlLoader();
//构建 VrmlLoader 对象
Scene s=null;
S=l.load(filename);
//导入*.wrl 文件
filename="*.wrl"
objRoot.addChild(s.getSceneGroup());
//将三维形体添加到 objRoot
}

```

上述代码说明了在 Java3D 场景中添加 VRML 造型场景的过程。先根据 Java3D 的场景图数据结构构建一个虚拟世界，然后将 TransformGroup 对象添加进去即可。

4 虚拟场景中的交互操作

4.1 立体仓库的调用

本系统采用 JDBC(Java Database Connectivity)作为数据引擎，建立三维虚拟场景交互系统，使用户的交互动作得到后台数据的支持。

该系统包含两个区域：虚拟场景和控制区。虚拟场景区域主要显示三维动画、虚拟漫游等操作；在控制区，通过对参数的设置和改变来控制三维动画。

在入库操作中，主要用到自动寻址和路径优化两种算法。自动寻址算法就是在预先没有设定货物存放位置时，堆垛机将按自己定义的入库规则，在立体货架上查找离出货台最近且货物为空的位置，并将货物放入。路径优化算法是在位置参数已经设定给堆垛机赋值的一种算法，使堆垛机能够在最短的时间内，经过最短的路径将货物入库。

4.2 物流仿真

物流系统的仿真是典型的离散事件系统仿真，其核心是时钟推进和事件调度机制^[4]。离散事件系统是指系统状态在某些随机时间点上发生离散变化的系统。该系统仿真设计主要采用仿真时钟和事件调度法来实现出库、入库、掉库等交互操作，并带有一定的随机性。

4.2.1 仿真时钟

仿真钟用于表示仿真时间的变化。在离散事件系统仿真中，由于系统状态变化是不连续的，在相邻两个事件发生之前，系统状态不发生变化，因此仿真钟可以跨越这些“不活动”周期，从一个事件的发生时刻，推进到下一个事件的发生时刻。

由于仿真是对系统状态在一定时间序列内的动态描述，因此仿真钟是仿真的主要自变量。仿真钟的推进方法有 3 类：事件调度法，固定增量推进法和主导时钟推进法。其中，仿真钟所显示的是系统仿真所花费的时间，与计算机运行仿真模型的时间度量是不同的。

4.2.2 事件调度法

事件调度法是面向事件的方法。通过定义事件，按时间顺序处理所发生的一系列事件。记录每个事件发生时引起的系统状态变化，从而完成系统的整个动态过程仿真。由于事件都是预定的，状态变化发生在明确的预定时刻，因此这种方法适合于活动持续时间比较确定的系统。

在事件调度法中，仿真钟是按下一时间步长法来推进的。

通过建立事件表，将预定的事件按时间发生的先后顺序放入事件表中。仿真钟始终推进到最早发生的时刻。

然后处理该事件发生时的系统状态变化，进行用户所需要的统计计算。

4.3 虚拟场景漫游

4.3.1 场景漫游

场景漫游是为了达到在场景中自由行走的效果，可以以不同的角度观察场景中的三维形体。这里，通过两条途径来实现：(1)改变观察者的视点；(2)改变坐标系的位置。在 Java3D 中，通过对抽象类 Behavior 的激励来达到场景漫游的目的。

Behavior 类的运行机制是：程序刚开始运行，向场景中安装特定的 Behavior 对象，该对象接收特定的激励(如键盘按键、鼠标移动、一个定时器消息)，然后对这一激励作出响应(如改变观察者的视角及物体位置、颜色)。

对于观察者的移动，Java 3D 提供了一个 KeyNavigator Behavior 类，该类通过响应键盘的输入来控制观察者的移动。Java 3D 也提供鼠标响应的 MouseBehavior 类，可以根据不同的鼠标输入来平移、旋转和缩放物体，从而达到交互漫游的效果。

4.3.2 漫游控制

实现漫游控制的步骤有：创建新场景，读入模型，将模型按正确的位置摆放，设置视点的位置、方向、视角，生成显示图像，初始化鼠标，将其与视点相连，对路径信息进行插值，并进行路径回放，实现固定路径漫游。

在漫游系统中，漫游方式的控制是通过相机采用 3 种控制方式实现的^[5]：(1)静态控制。进入漫游系统时，初始视点是通过相机的直接控制实现的，将相机的位置和旋转角度设定在满意的地方即可。(2)旋转控制。当需要环绕某物体仔细观察时，可以设定相机以该物体为圆心做圆周运动，并使相机始终朝向这一圆心，这也是对相机的直接控制。(3)跟踪控制。相机跟踪某运动物体时，观察角度随物体的运动而改变，它与跟踪场景中某一点的原理类似，只不过此点是运动的。

5 系统设计实例

系统由 3 个部分组成：(1)菜单栏；(2)快捷按钮区；(3)显示主界面。主界面又由 2 个部分组成：控制区和虚拟场景区。可以根据菜单栏对系统做相应的操作，如：定单管理，客户管理，库存查询，场景设置。快捷键主要是对显示主界面的控制，如：单击出库操作，在控制区输入出库货物的位置，查询数据库中是否存在货物。如果货物存在，执行出库操作，右边三维显示界面就做相应的出库动画，用于仿真物体出库；如果货物不存在，说明该位置没有货物，要重新输入。另外，系统提供了虚拟物流实验室基本构造元。可以根据不同的实验目标和要求，动态地组合构造元，进行仿真和交互实验。

6 结束语

物流是一门实践性很强的学科，伴随物流教育的发展，在全国出现了物流实验室建设的热潮。然而，昂贵的物流设备以及实验室庞大的占地面积使国内多数高等院校难以承担。因此，研究虚拟物流系统具有重大意义，并具有很好的应用前景。

本文从虚拟环境的构建、三维图形的优化以及人机交互
(下转第 264 页)