

# 基于 Java 3D 的交互式三维动画的研究

王静秋, 王国忠

(南京航空航天大学 机电学院, 江苏 南京 210016)

**摘要:**为了建立基于工程图学的虚拟实验室,在计算机的屏幕上虚拟现实零件的装配,笔者通过对两种虚拟语言 VRML 与 Java 3D 的分析比较,选定以 Java 3D 为虚拟平台,在引入 .obj 文件的基础上,按照现实中零件的安装和工作过程,通过设定相应的程序来实现零件的交互和装配过程,并完成了对零件交互式三维装配动画以及工作原理的论述。以压板为例,给出了一个基于 Java 3D 的交互式三维装配动画的例子,实现了该装配体在虚拟现实中的要求,达到了虚拟装配的目的。从而验证了该方法在现实零件装配中的预见性和优越性。

**关键词:**VRML; Java 3D; 虚拟装配

中图分类号:TP312

文献标识码:A

文章编号:1673-629X(2011)09-0148-05

## Research of Interactive 3D Animation Based on Java 3D

WANG Jing-qiu, WANG Guo-zhong

(College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics  
and Astronautics, Nanjing 210016, China)

**Abstract:** The purpose of this research is to establish virtual laboratory which is based on engineering graphics and to realize virtual assembly of parts in the computer screen. Java 3D was chosen for virtual platform compared with VRML language. In accordance with the reality of the parts of the installation and work processes based on introducing .obj file, realized the interaction and assembly of parts by setting button program and completed the description of interactive 3D assembly animation. Finally by taking plank for example, gave an interactive 3D assembly animation, realized virtual assembly and met the requirements in virtual reality. Proved the predictability and superiority of this method in the real component assembly.

**Key words:** VRML; Java 3D; virtual assembly

## 0 引言

WWW(World Wide Web)为3D图形增添了一种有趣的新方法,利用3D图形可以创建完全存在于计算机内存的虚幻世界,也就是被大众媒体称作“虚拟现实”<sup>[1]</sup>。三维图形技术的鼻祖是SGI公司推出的OpenGL三维图形库,而在此基础上产生了两种重要的虚拟语言VRML(Virtual Reality Modeling Language)和Java 3D,由于它们具有跨平台性、面向对象性与当前网络技术相结合的特点,使虚拟语言很快得到人们的认可并且在各行各业得到迅猛的发展。

笔者通过对VRML和Java 3D在设计交互式三维动画中效果的比较,确立了以Java 3D作为设计工具,在Pro/E、3DS MAX建立三维零件模型并获得静态文件的基础上,将静态文件导入Java 3D中,利用编程代

码实现了交互式装配动画。通过探讨虚拟世界中零件装配的过程,得出了实现装配体缩放、任意角度旋转、平移、初始状态转换以及在任意时刻的运动与停止的方法,最后给出了一个交互式三维零件装配的实例。

## 1 VRML 和 Java 3D

### 1.1 VRML

VRML同HTML语言一样,是一种ASCII的描述性语言<sup>[2]</sup>,VRML也是通用的3D数据描述语言<sup>[3]</sup>。VRML文件的基本要素包括文件头、造型、原型、事件和路由。

VRML使用场景图(Scene Graph)数据结构来建立3D场景。场景中节点(Node)是描述对象及其属性的重要组成部分,节点按照一定的规则构成场景图,尤其是其中的Script节点对VRML提供了强有力的补充,通过Script节点可以利用Java或者JavaScript程序脚本可以满足开发者更多的要求,大大地扩展了VRML的功能,广泛地应用于科学研究、教育、工程、建筑、电子商务等领域。

收稿日期:2011-02-23;修回日期:2011-06-18

基金项目:南京航空航天大学教改项目(V0956-051)

作者简介:王静秋(1972-),女,副教授,主要研究领域为计算机图形图像处理、计算机辅助工业设计。

## 1.2 Java 3D

Java 3D 被称为第 4 代的三维图形应用程序接口<sup>[4]</sup>,Java 3D 是 Java 2 SDK 的标准的扩展,它可以和普通的 Java 2D、Swing、AWT 很好地结合,并且对底层的图形库 OpenGL 和 DirectX 进行了封装,其意义非同寻常<sup>[5]</sup>。Java 3D 的优点在于:提高了编写三维图形程序的层次,它使用户只考虑图形对象本身,而不用自己去实现这个对象,也不需要考虑光照、着色、碰撞检查等极其复杂的图形学问题;Java 3D 集成了 Java API 的功能,一次编程就可以跨平台运行,而且可以充分借用 Java 的各种功能;Java 3D 最突出的优点在于它代码的可传输性,它的代码可以自由传输,能够使 applet 方便地从服务器传给客户端,也就是说传输的不是图像本身,而是控制三维图像生成的图像和程序,因而大大节省了网路传输的数据量<sup>[6]</sup>。

## 1.3 VRML 与 Java 3D 在零件装配上的比较

Java 3D 和 VRML 作为两种虚拟现实的语言,都可以动态或者静态的显示三维图像,并实现交互。本课题的模型都通过 Pro/E 建模,将模型引入 Java 3D 和 VRML 中进行动态显示的图像分别如图 1、图 2 所示。

通过比较,可以发现 VRML 在该课题中的诸多不足:

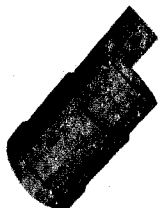


图 1 通过 VRML 编程获得的零件图像



图 2 通过 Java 3D 编程获得的零件图像

首先从技术上在 Pro/E 中的建模需要导入到 3DS MAX 进行转换才可以被 VRML 利用;虽然 VRML 编程简便,但是与之相适应的浏览器性能不佳;VRML 中的缺少了有效的碰撞检测的节点;最重要的是在 VRML 中,每个零件装配的临界点不能准确控制,随时控制零件装配的动画和状态的复原更是一个难关。

其次通过两个图像的对比读者可以观察到:通过 VRML 编程得到的零件图像表面粗糙;拉近 VRML 获得的图像,因为视角的原因会让观察者产生“零件畸形”的感觉。

再次 VRML 语言内插器节点计算功能薄弱,对于基于物理的动画缺乏描述能力,再加上其造型和动画都有一定的随意性,在动态仿真、科学计算可视化、虚拟装配等要求精确复杂控制的领域表现得不尽人意,限制了 VRML 在虚拟装配中的应用<sup>[6]</sup>。

利用 Java 3D 编程能够克服在上面的诸多缺点,并且 Java 3D 在检测碰撞方面有良好的效果。不同的检测算法具有不同的特点和面向不同的应用对象<sup>[7,8]</sup>,其中 Lin Canny 最近特征算法<sup>[9]</sup>与基于层次数抓结构的层次算法<sup>[10]</sup>在碰撞检测方面达到良好的效果,由此笔者选定 Java 3D 作为本次课题的工具。

## 2 利用 Java 3D 设计交互式三维零件装配动画

以虚拟语言 Java 3D 为编程工具,通过编程对模型的控制,实现了装配体的缩放、任意角度的旋转、平移及在任意的时刻的交互以及机构的工作原理。该设计分为系统结构设计和程序设计两部分。

### 2.1 系统结构设计

Java 3D 的编程空间采用的是场景图结构(Scene Graph Structure),它是有向无环图,它的场景图结构<sup>[5]</sup>和设计中的逻辑结构图分别如图 3、图 4 所示。

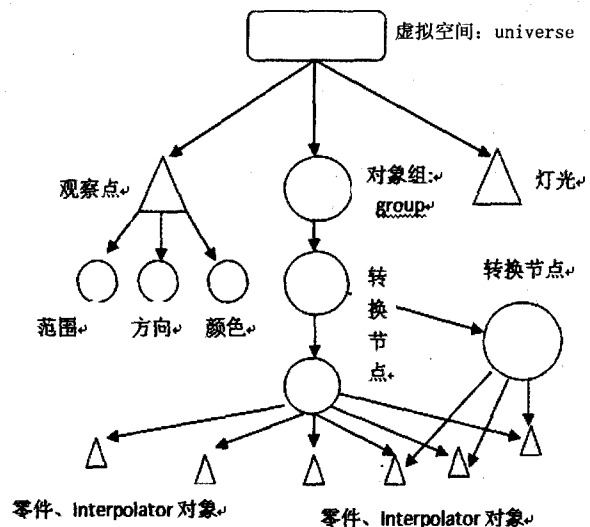


图 3 场景图结构

Java 3D 应用程序的结构和一棵大树很相似,从逻辑结构图可以看出每个场景图中都有一个虚拟空间形象的称为“根节点”,在根节点的下面对应一个场所,每个场所有多个基准坐标系,每个坐标系里面摆放着三维形体,这就像一个树干上分散着很多树枝,树枝上面又长满了树叶一样。

### 2.2 程序设计

#### 2.2.1 背景、材质、灯光的设计

按照图 4,笔者把背景、材质的颜色、灯光以及各

种对象的作用范围放在根节点上,由于三维零件的装配,对灯光、颜色、材质、贴图没有特殊的要求,只需对光线方向、环境光与定向光进行合适的设置。

VirtualUniverse: 虚拟空间

Locale: 场所

BranchGroup: 分支节点

TransformGroup: 转换节点

Shape3D: 三维形体

Appearance: 外观

Geometry: 几何图形

ViewPlatform: 观察平台

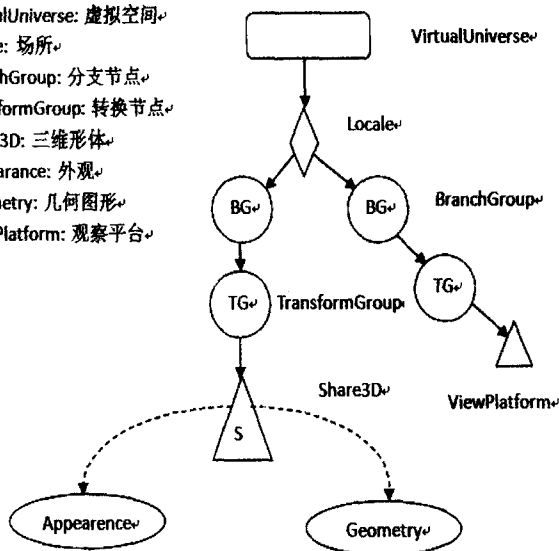


图4 设计中的逻辑结

### 2.2.2 实现交互的程序设计

实现装配过程的交互,首先要响应键盘和鼠标的操作,在 Java 3D 的工具类中提供了通过对鼠标的输入进行交互的 MouseBehavior 类,通过响应鼠标的左键、中键和右键分别实现物体的旋转、缩放和平移,逻辑图中所有的物体都被封装在 TransformGroup 节点中,要实现整个装配过程中的旋转、缩放,在允许 TransformGroup 对象有读写能力的基础上,还要设置 MouseBehavior 对象要作用于 TransformGroup 对象。

实现平移的代码如下:

```
MouseZoom behavior = new MouseZoom ( worldTrans );
//创建 Mousebehavior 对象,并作用于 TransformGroup 对象
(worldTrans)
behavior.setSchedulingBounds( bounds );
//设置对象的作用范围
sceneRoot.addChild( behavior );
//将对象添加到场景中
```

### 2.2.3 .obj 文件的引入

Java 3D 可构造出一些漂亮的几何形体,但要想通过程序来构造复杂的几何形体,其困难程度是可想而知的<sup>[11]</sup>。Java 3D 提供了一组可以用来导入一些常用的 3D 图形文件接口,可以调用其它格式的三维图形文件获得复杂的几何形体。笔者在 Pro/E 中建模,导出 .obj 文件,通过 Java 3D 调用 .obj 文件获得零件对象,这些对象放在分别放在各自的局部坐标系 TransformGroup 中,这些局部坐标系最终放在一个共同的坐标系内。

下面给出了引入 .obj 文件的程序:

```
public BranchGroup createSceneGraph( String filename );
.....
int flags = ObjectFile.RESIZE;
ObjectFile f = new ObjectFile( flags, (float)(49.0 * Math.
PI / 180.0));
Scene s = null;
try { s = f.load( filename );
.....//错误处理
objTG.addChild( s.getSceneGroup( ) );
public objLoader( String filename );
filename = "taotong.obj";
.....
new MainFrame( new objLoader( filename ) );
```

### 2.2.4 零件装配动画的设计

实现零件的装配动画,Java 3D 为编程者提供了 Interpolator 内插值器对象和 Alpha 对象,通过这两者的结合可以编写出针对三维物体的运动和外观的可视化效果的变化。

怎么解决三维物体的运动和外观是一个重要的内容,由于每个零件都放在独立的坐标系中,实现物体运动其实就是实现零件所在坐标系的运动,因此首先要确定好每个零件在动画编程中所需的时间控制环节,也就是要正确配置 Alpha 对象,根据不同零件的装配要求可以选择 PositionInterpolator 对象(使局部坐标系在两个点之间按照 Alpha 提供的方式移动位置)、RotationInterpolator 对象(绕某轴线转)、PositionPathInterpolator 对象(在规定的周期内,对某一个坐标系按照指定的路径制定的运动方式进行坐标系的移动)或者其它对象,确定好每个零件的运动方式、运动时间、运动路线以实现针对不同零件的运动控制;由于在 Java 3D 中笔者引入了所有需要装配的零件,这就造成了在零件装配的开始,最后装配的零件就会一直显示在页面,给人们造成视觉的瑕疵,为了解决这个问题,可以在某个零件装配之前,先把该零件设置为背景色或者把零件设置为透明色,让它处于“不显示”状态,当需要该零件装配的时候,再将零件显示出来进行装配,也就是要对零件(也就是零件所在的坐标系)进行可视化效果的设置,可以通过 Alpha 对象和形体颜色 ColorInterpolator 对象或者形体透明度 TransparencyInterpolator 对象来改变形体效果获得想要的结果。需要强调的是 Interpolator 内插值器对象作用的坐标系必须要允许写的功能:

```
objMovePos.setCapability( TransformGroup.ALLOW_TRANSFORM_WRITE );
posInt = new PositionInterpolator( alpha, objMovePos );
在用 ColorInterpolator 对象时,为了使颜色发生变化,需要在定义形体材质时通过下面设定允许颜色产
```

生变化<sup>[12]</sup>:

```
Material mat = new Material();
mat.setCapability( Material.ALLOW_COMPONENT_WRITE);
```

### 2.2.5 按钮程序的设计

在装配过程中,除了上面与鼠标进行交互外,还要控制任意时刻装配过程的运动与暂停;当把装配体(坐标系)旋转任意角度后,还要考虑能够使装配体(坐标系)快速回到初始的角度(缺省状态),这两个是在交互过程中需要解决的重要问题。Java 3D 提供了 ActionListener 监听器用来监听各种 AWT 事件。

论文解决这两个问题的思路是:设定功能按钮,通过监听器监听这个功能按钮是否被按下,如果指定的按钮按下,监听器发出按钮按下的事件程序则通过处理事件的 actionPerformed 方法进行事件的处理<sup>[9]</sup>,转到设定好的处理程序,达到控制的目的。由于是控制零件(坐标系)的运动,就是控制 Interpolator 内插值器对象和 Alpha 对象,先前在每个零件的坐标系里面都定义了 Interpolator 内插值器对象和 Alpha 对象,在“stop”和“move”按钮的程序里面要合理地设定所有 Interpolator 内插值器对象的 setEnable 方法以及 Alpha 对象的 resume 方法,一定要保证按下“stop”按钮的时候 Alpha 对象一定暂停产生 Alpha 值!对于初始状态的设置,在定义根节点的子节点的时候,将其子节点下的 Shape3D(三维形体)保存起来以便于后面的调用。

按钮设定的程序如下:

```
trBackup = new Transform3D( tr );//保存原始的 Transform3D
以便于 normal 按钮使用
```

```
public void actionPerformed( ActionEvent e )
{
    long time1 = 1;
    if ( e.getSource() == activateB ) {
        alpha1.pause();
        .....
        posInt1.setEnable( false );
        .....
    }
    if ( e.getSource() == normalB ) {
        tr.set( trBackup );
        worldTrans.setTransform( tr );
    }
    if ( e.getSource() == rotateBX ) {
        if ( alpha1.isPaused() ) {
            alpha1.resume();
            .....
        }
        posInt1.setEnable( true );
        .....
    }
}
```

### 2.2.6 机构原理动画的设计

在装配体完成的最后,可以给出装配体的工作原理,不同的机构对应各自的工作原理,这也是一个动画设计的过程,与上面的动画不同的是机构工作原理可

能是几个零件的组合,这个组合成为一个整体进行旋转和平移,这就要求把几个零件所在的坐标系添加到一个共同的坐标系中,以这个坐标系为对象,把这个对象看作一个零件,通过 Interpolator 内插值器对象和 Alpha 对象的结合可以编写出针对几个对象的不同的运动,这样便形成机构的工作原理动画。可以通过设置按钮控制该动画,由于机构工作原理的动画是在装配完成之后进行的,这就要求将 Interpolator 内插值器对象作用于整体坐标系,并且将 setEnable 方法设置为 false,并且对应的 Alpha 对象不能产生 Alpha 值,只有在按下按钮的时候才将 Interpolator 对象和 Alpha 对象激活。

通过上面的系统结构设计和程序设计两个过程,可以实现零件的装配、机构工作原理的动画和交互过程,程序设计是本次课题的关键,尤其是多个 .obj 文件的引入、零件的运动轨迹以及按钮程序中 Interpolator 内插值器对象和 Alpha 对象的有效结合是课题解决的重点。

## 3 一个关于交互式三维零件装配动画实例

下面给出了压板的装配动画(图5是对应的效果图),压板是由八个零件组成,先设定根坐标系 SceneRoot,对应的子坐标系 WorldTrans,在子坐标系 WorldTrans 里面放置了灯光,添加了与鼠标的交互的

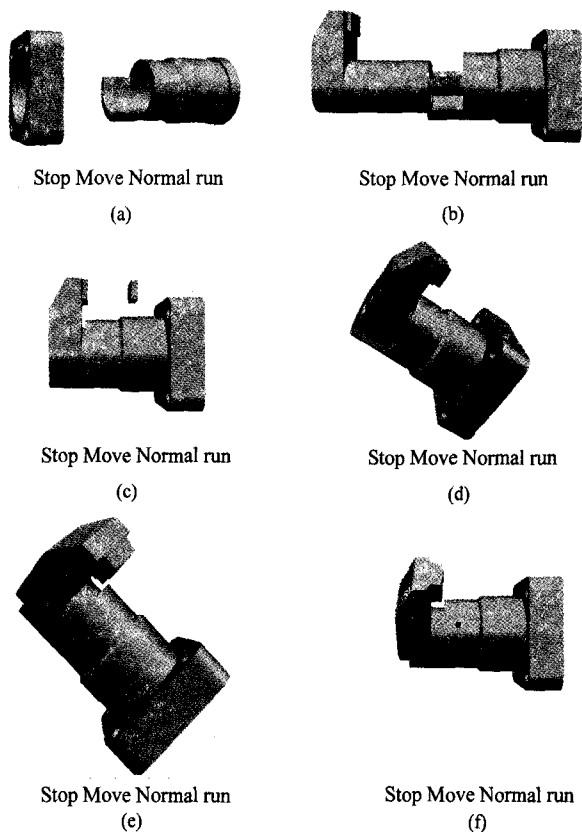


图5 效果图

MouseBehavior 类,并引入了九个子坐标系,八个子坐标系里面分别引入八个零件,另一个子坐标系里面组合几个零件,每个子坐标系都有作用于坐标系的 Interpolator 内插值器对象用以实现动画过程,最后又设计了几个按钮的程序。

#### 4 结束语

通过对 Java3D 场景图分析,给出了实现三维动画的逻辑结构图,讨论了文件的引入、零件的装配和交互方法以及按钮的交互程序。最后通过一个实例实现了交互式三维零件装配的动画、交互过程,形象地表达了机构的工作原理,满足了方便、形象、直观的要求。用虚拟的计算机世界来表达了现实的空间,可以广泛地应用于教学、科研、网页建设等领域,有广阔的应用前景。

#### 参考文献:

- [1] Marrin C, Campbell B. 21 天学通 VRML2 [M]. 王海燕,李庆军,曲庆尧,等译. 北京:人民邮电出版社,1998:1-3.
- [2] 吴小华,姜安德,周玲. VRML 从入门到精通 [M]. 北京:国防工业出版社,2002:6-9.
- [3] Lea R, Matsuda K, Migashita K. 妙用 Java 及 VRML 开发三维图形 [M]. 葛晓程,译. 北京:电子工业出版社,1995.

(上接第 147 页)

在中风诊断、治疗、用药中的应用问题。在以中风方剂、症状、药味为基础的数据库平台上,依据本次挖掘的目的,发现了一些药物间配伍的经验及症状和药味间的关联关系。由于目前中医具有对症状描述的模糊性及不完全性等特点,给数据挖掘增加了难度<sup>[14]</sup>。随着当今中医标准化的研究,相信中医数据库的信息将更加一致,数据挖掘将在中医药现代化研究中发挥越来越重要的作用。

#### 参考文献:

- [1] 王越,桂袁义. 基于关联分析的数据挖掘在体检 CRM 中的应用 [J]. 重庆理工大学学报(自然科学),2010,24(3):37-42.
- [2] 文拯,梁建武,陈英. 关联规则算法的研究 [J]. 计算机技术与发展,2009,19(5):56-58.
- [3] 李景文,刘军锋,闫遂军,等. 一种改进的最大频繁集发现算法 [J]. 计算机技术与发展,2008,18(10):113-119.
- [4] 邱祝芳,焦贤龙,高升译. 数据挖掘原理与应用 [M]. 北京:清华大学出版社,2007.
- [5] McGinnis S, Madden T L. BLAST: at the core of a powerful and diverse set of sequence analysis tools [J]. Nucleic Acids Res,2004,32:20-25.

- [4] The Fourth Generation of 3D Graphics APIs Has Arrived [EB/OL]. 1987. <http://javasun.com/products/Java-media/3D/collateral/>.
- [5] 都志辉,刘鹏,陈渝. Java3D 编程实践——网络上的三维动画 [M]. 北京:清华大学出版社,2002:1-11.
- [6] 张本生,刘海光,黄波. 基于 VRML 和 Java 的虚拟装配复杂控制的实现 [J]. 机械工程与自动化,2010(1):24-56.
- [7] Lin M C. Efficient Collision Detection for Animation and Robotics [M]. Department of Electrical Engineering and Computer Sciences, University of California Berkeley, 1993.
- [8] 王志强,洪嘉振,杨辉. 碰撞检测问题研究综述 [J]. 软件学报,1999,10(5):545-551.
- [9] Lin M C, Canny J F. A fast algorithm for incremental distance calculation [C]//In: Proceedings of the IEEE International Conference on Robotics and Automation. Sacramento, CA; [s. n.], 1991:1008-1014.
- [10] Guibas L J, Hsu D, Zhang Li. A hierarchical method to real time distance computation among moving convex bodies [J]. Computational Geometry: Theory and Applications, Special Issue on Virtual Reality, 2000,15(1-3):51-68.
- [11] 余小燕. 基于 Java 3D 的交互式三维动画编程技术 [J]. 工矿自动化,2003(6):57-59.
- [12] 张杰. Java 3D 交互式三维编程 [M]. 北京:人民邮电出版社,1999:197-198.

- [6] 宋雨,赵建利,王保义. 关联规则挖掘中最大频繁集的双向查找算法 [J]. 华北电力大学学报,2005,32(2):67-71.
- [7] Agrawal R, Imielinski T, Swami A. Mining Association Rules between Sets of Items in Large Databases [C]//Proceedings of the 1993 ACM SIGMOD Conference. Washington D C; [s. n.], 1993:207-216.
- [8] 谢邦昌. 商务智能与数据挖掘 [M]. 北京:机械工业出版社,2008.
- [9] 陈玉婷,王斌,刘博,等. 关联规则挖掘算法介绍 [J]. 计算机技术与发展,2006,16(5):21-25.
- [10] 王爱平,王占凤,陶嗣干,等. 数据挖掘中常用关联规则挖掘算法 [J]. 计算机技术与发展,2010,20(4):105-108.
- [11] 邹志文,朱金伟. 数据挖掘算法研究与综述 [J]. 计算机工程与设计,2005,26(9):2304-2307.
- [12] Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation [C]//In: Proceeding of the 2000 ACM SIGMOD Conference on Management of Data, Dallas, TX; [s. n.], 2000.
- [13] 孙吉贵,刘杰,赵连宇. 聚类算法研究 [J]. 软件学报,2008,19(1):48-61.
- [14] 姜兆顺,倪青,林兰. 数据挖掘技术在中医药研究中的应用 [J]. 中华现代内科学杂志,2006,3(11):1240-1242.