doi: 10.7690/bgzdh.2018.02.016

Java 3D 虚拟现实技术在水下发射模拟中的应用

洪华军，许统华，吴建波
（中国船舶科学研究中心，江苏 无锡 214082）

摘要：为实现集成试验模型仿真，动态轨迹模拟功能，构建基于 J2EE 的水下发射综合试验数据管理平台。使用 Java 3D 虚拟现实技术，对水下发射模型应用进行分析。首先介绍了 Java 3D 技术，对其优势进行分析，并阐述了水下发射 3 维显示模型和弹道姿态动态复现模拟相关知识。采用 Java 3D 交互与动画技术，构建试验模型位移和偏转变化的动画，研究各种工况下的试验模型姿态变化曲线；将 Java 3D 虚拟现实技术应用于水下发射中的虚拟模型 3 维显示和弹道姿态动态复现模拟。结果表明：Java 3D 技术使试验模型显示更直观、清晰，更有利于试验数据分析。

关键词：Java 3D；弹道；虚拟现实
中图分类号：TJ761.3　　**文献标志码：**A

Application of Underwater Launch Simulation in Java 3D Virtual Reality Technology

Hong Huajun, Xu Tonghua, Wu Jianbo
(*China Ship Science Research Center*, *Wuxi* 214082, *China*)

Abstract: For realizing integrated test model simulation and dynamic orbit simulation function, establish underwater launch comprehensive test data management platform based on J2EE. Use Java 3D virtual reality technology, analyze the underwater launching model application. At first introduce the Java 3D technology, analyze its advantages, and introduce the underwater launching 3D display model and ballistic gesture dynamic recurrent simulation related technology. Use Java 3D interaction and animation technology to establish the test model displacement and deflection change animation and research test model gesture change curve under varied work situation. Apply the Java 3D virtual technology in virtual model 3D display and ballistic gesture dynamic recurrent simulation of underwater launching. The results show that the Java 3D technology can make the test model more intuitively, clearly and helpful to the test data analysis.

Keywords: Java 3D; ballistic; virtual reality

0　引言

水下发射是从潜艇以及其他装置发射航行体的方式和技术[1]。水下发射最大优点是隐蔽性好、生存能力强，但技术比较复杂，世界各军事强国对水下发射技术都十分重视。构建基于 J2EE 的水下发射综合试验数据管理平台，需要集成试验模型仿真，动态轨迹模拟功能。Java 3D 是重要的仿真技术之一，能很好地满足当前系统这一要求。 利用 Java 3D 提供的 API，可以编写出 3 维动画、各种计算机辅助设计和 3 维仿真等。利用 Java 3D 编写的应用程序，只需调用这些接口进行编程，在客户端通过 Java 虚拟机就可以访问。

笔者使用 Java 3D 技术的优势，对水下发射模型应用进行了分析，将 Java 3D 虚拟现实技术应用于水下发射中的虚拟模型 3 维显示和动态模拟 2 方面。

1　Java 3D 虚拟现实技术

1.1　Java 3D 技术

Java 3D 使用 Java 实现基于客户端的交互式 3 维图形接口，是 Java 语言在 3 维领域的扩展。针对 Java 3D 存在处理用户交互事件速度慢和难以显示大量对象的缺陷，通过优化场景图的组织结构和数据结构得到很大的改善，良好的场景图组织和数据结构使显示的速度和对象数量明显提高[2]。

Java 3D 场景图(scene graph object)由不同类对象的节点对象构成，形成一个树形结构。其中，根结点是虚拟宇宙结点(virtual universe)，其他子结点从根结点一直往下延伸，不能形成回路，如图 1[3]。

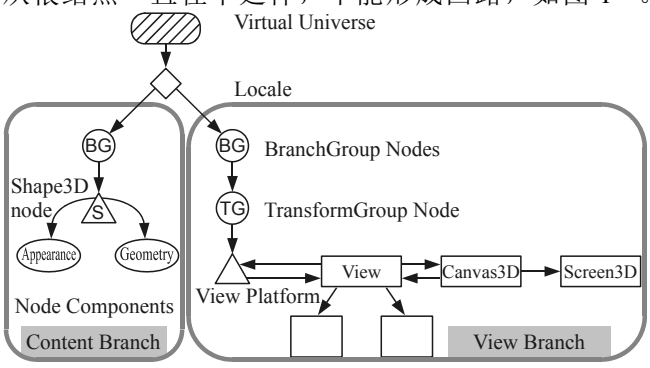


图 1　Java 3D 场景

Java 3D 虚拟宇宙(virtual universe)是一个由

收稿日期：2017-11-13；修回日期：2017-12-16
作者简介：洪华军(1984－)，男，浙江人，硕士，从事试验数据管理、企业信息化研究。

Java 3D 类的实例构成的场景图，场景图分为 2 大类：视图分支图(view branch)和内容分支图(content branch)。视图分支子图指定了视图参数比如观看位置和方向；内容分支图指定了虚拟世界的内容：图形、表面、位置、动作、声音以及光线等。

1.2　Java 3D 的优势

Java 3D 在科学计算、虚拟现实和仿真技术领域都有广泛的应用，主要原因在于：

1) 以 Java 为基础，继承了 Java 的所有优点，具有优秀的开放性、扩展性和可维护性。

2) 作为第 4 代 3D 图形 API，借鉴了 VRML 虚拟现实思想，简单易上手。只需图形学功底，开发者便能开发出专业的动画、仿真等产品[4]。

3) 功能强大，高层开发。由于 Java 3D 在底层借助了 Open/GL、DirectX 强有力的支持；在高层，开发者不再需要负责对象渲染、碰撞检查的编程任务[4]。

4) 基于场景图结构。这种树型层次结构，既有利于描述现实中的对象，又有助于计算机实现。

5) 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，即传输的不是图像本身，而是控制 3 维图像生成的图像和程序，因此大大节省网路传输的数据量[5-6]。

6) Java 3D 使用硬件加速技术实现速度优化。Java 3D底层通过 DirectX 或 OpenGL 实现 3D 硬件加速，并使用视锥体消除法处理技术，同时采用多线程，实现速度的优化[7]。

2　水下发射

2.1　模型 3 维显示

模型基本为圆柱形旋转体，采用半球头的圆柱体代替。约定坐标：参考系为弹体坐标系，采用柱坐标系(z, a, r)，第 1 项 z 为传感器到头顶平面的距离，第 2 项 a 为周向位置角度，第 3 项 r 为传感器到中心轴的距离。周向位置方向约定：从头顶向尾部看去，以逆时针方向，背水面端面为Ⅰ象限(0°)，侧面 1 为Ⅱ象限(90°)、迎水面端面为Ⅲ象限(180°)、侧面 2 为Ⅳ象限(270°)。

例如：模型头部中心位置，按照约定即等于(0, 0, 0)；模型底部中心位置：(L, 0, 0)(L=尾部距离头部的距离)；需要说明的还有Ⅰ偏Ⅱ45°，Ⅱ偏Ⅲ45°，Ⅲ偏Ⅳ45°，Ⅳ偏Ⅰ45° 即=45°, 35°, 225°, 315°。演示模型的 CAD 如图 2。

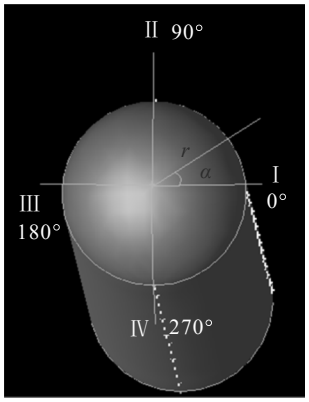


图 2　模型 CAD

2.2　弹道姿态动态复现模拟

在模型运动过程中，弹道姿态会发生变化，弹道包括 X, Y, Z 方向的位移，3 个方向的速度、加速度和角速度等。指定弹道物理量按指定变量(一般为时间或者 Y 方向位移)的曲线。需要观测试验模型的俯仰姿态，输出定性全局动画[8]。

3　JAVA 3D 在水下发射模拟中的应用

为了实现系统总体集成，将试验数据、试验模型和试验仿真都集成到 B/S 结构的 J2EE 综合管理平台下，采用 J2SE 构建模型 3 维显示，弹道姿态动态模拟应用程序采用 JNLP 方式在客户端加载应用。一个 JNLP 客户端是一个应用程序或服务，可以从宿主于网络的资源中加载应用程序。

3.1　模型 3 维显示

按照 Java 3D 构建 3 维模型步骤[9]，首先创建一个 Canvas 3D 对象：

```
Canvas3D c = createUniverse();
```

然后创建一个 VirtualUniverse 对象，

```
SimpleUniverse univ = new SimpleUniverse(c);
```

构建一个视图子图：

```
BranchGroup scene = getBranchGroup();
```

其中在方法 getBranchGroup()通过创建 View，ViewPlatform，PhysicalBody，PhysicalEnvironment，构建出整个 3 维模型。

模型参数，信号定义数据通过 JNLP 文件中 URL 传参形式传入，为模型生成准备数据。JNLP 文件内容如下：

```
<?xml version="1.0" encoding="GBK"?>
<jnlp spec="1.0+" codebase="
http://192.168.3.168:8888/Module3D" href="">
<information>
  <title>ImageJnlpServlet</title>
  <vendor>IT8 Tech</vendor>
```

```
    <homepage href="index.html"/>
    <description>Web Start Version</description>
    <description kind="short">5.0</description>
  </information>
  <security><all-permissions/></security>
  <resources>
    <j2se version="1.5+"/>
    <jar
href="imageMeasureJnlp/ModuleShowJava 3D.jar"/>
  </resources>
  <application-desc
    main-class="Java 3DModuleShow">
  <argument> 30,5,5</argument>
  <argument>0,0,0,0,0;1,47,2.5,1,2;1,134,2.5,1,7;4,
47,2.5,1,3.7;……
  </argument>
  </application-desc>
  </jnlp>
```

数据格式中参数 30, 5, 5 表示模型缩比后的长、宽、高，设为 $L(l, w, h)$，其中长度为这个试验模型的长度，包括半圆球的半径。宽和高是相等的，为半圆球和圆柱体的直径。这 3 个参数用于构建整个 3 维模型。

创建 2 个几何体：

球体：Sphere sphere=new Sphere(w/2, -1, 80);

圆柱体：Cylinder cylinder=new Cylinder(w/2,(l - w/2))。

分别加入到对应的场景图中，并设置其属性，呈现如图 3 所示模型。

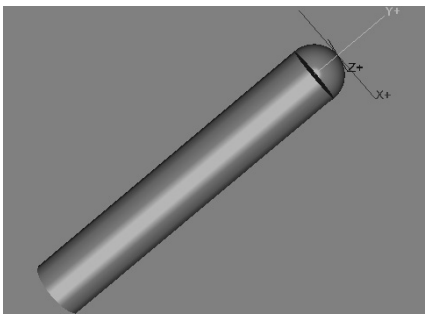


图 3 模型

参数 0,0,0,0,0; 1,47,2.5,1,2; 1,134,2.5,1,7; 4,47,2.5, 1,3.7;…是信号定义数据参数，各个测点数据通过分号隔开，每个测点数据有 5 个值，定义为 $P(z, a, r,$ type, pa)，其中 z,a,r 表示测点的位置。type 为信号类型，是枚举类，包括压力(用 1 表示)、弹道(用 2 表示)、温度(用 3 表示)和载荷(用 4 表示)。Pa 参数表示对于该测点在对应信号类型下的值。

对于每个模型上的信号定义位置，数据库提供的是柱坐标(z, a, r)数据，需要将它转化为直角坐标，以便在 Java 3D 空间模型中确定各个测点的空间位置。

设测点坐标为 $P(X, Y, Z)$，通过 a 和 r 一起用来计算 Z, X 轴坐标。转换方式如下：

$Y = -z$;

$X = r\times\cos(a)$;

$Z = r\times\sin(a)$。

具体实现：

1) 测点定位。

通过构建一个 TransformGroup 和与之对应的 Transform3D，并且设置该 Transform3D 的 Translation 为：

new Vector3D(r*Math.cos(a),-z,r*Math.sin(a)); 来确定测点位置。

2) 压力表示。

通过构建各测点的法线长度来表示测点压力大小。

压力大小半径表示：

```
float R =z+ Float.valueOf(Pa);
```

构建一条法线：

```
double vert[] =
{r*Math.cos(a),-z,r*Math.sin(a),
R*Math.cos(a),-z,R*Math.sin(a),};
```

3) 标注表示。

使用标注描述测点位置的 3 维坐标，构建 Text3D txt = new Text3D(Font3D, textString, Point3f);

其中 textString 等于测点的 3 维坐标串。

测点位置、压力标注表示如图 4、图 5 所示。

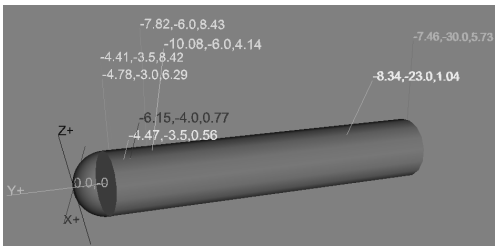


图 4 测点定位和压力标注图 1

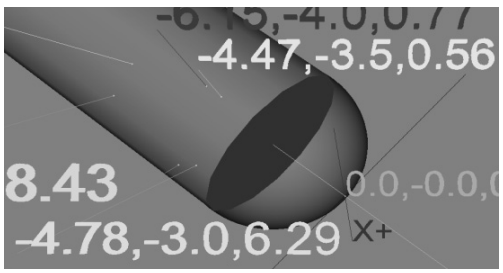


图 5 测点定位和压力标注图 2

3.2　弹道姿态动态复现模拟

试验模型在水下运动的过程中，在很短的时间内其姿态发生了变化，需要研究各种工况下的姿态变化曲线[10] 。笔者采用 Java 3D 交互与动画技术，构建试验模型位移和偏转变化的动画。

定义一个 Alpha 对象计时，该对象可以缩放周期大小。Interpolator 插入器使用 Alpha 对象来实施对场景图的更改。

Alpha xtranAlpha = new Alpha(-1, Alpha. INCREASING_ ENABLE, 0,0,10000,0,0,0,0,0); 该类型的 Alpha 使形体从左移到右，再将形体跳回到左边无限循环运行，上升时段时间设为 10 s。

由于模型在发射过程中不仅要实现平移运动，同时会沿 *X*,*Y*,*Z* 轴发生偏转运动，该运动形式正好与 Java 3D 中的 RotPosPatInterpolator 对象实现一致。

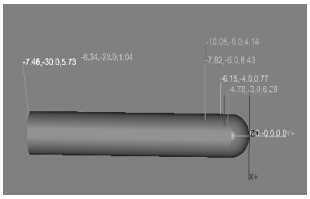
构建 3 个数组参数：

1) float knot[] = new float[pointCount]；指定各个递增时间节点 knot[0]=0, knot[pointCount-1]=1。

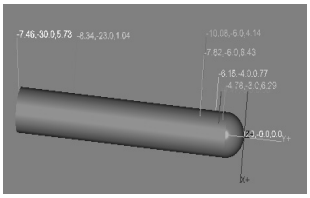
2) Point3f pos[] =new Point3f[pointCount]；指定形体在 knot 下的平移路径。

3) Quat4f quat[] =new Quat4f[pointCount]；用来定义形体的旋转方式。使形体在每一个 knots 数值之下都有一个旋转角度相对应。

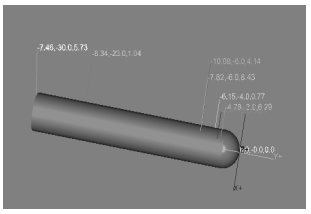
同模型 3 维显示一样，采用 JNLP 进行应用加载，3 个参数数组数据通过参数的形式传入。运行结果显示模型按照参数所指定的姿态在 3 维场景图中运动。图 6 为运行过程中的部分截图。



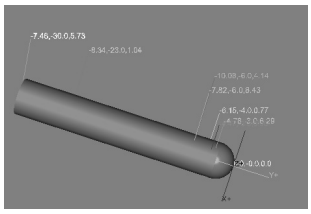
(a) 模型姿态变化过程 1



(b) 模型姿态变化过程 2



(c) 模型姿态变化过程 3



(d) 模型姿态变化过程 4

图 6　模型姿态变化过程

4　结束语

笔者通过 Java 3D 技术将水下发射航行体虚拟模型和动态复现模拟集成到试验数据综合管理平台，更有利于试验数据的直观表现和分析，对模型运动姿态变化规律的理解具有重要的意义。下一步将研究模型在运行过程中压力随时间变化情况。

参考文献：

[1] 姚昌仁, 张波. 火箭导弹发射装置设计[M]. 北京: 北京理工大学出版社, 1998: 1-20.

[2] 杨小冬, 胡立堂, 唐仲华. 基于 Java/Java 3D 的地层 3 维建模与可视化[J]. 测绘学报, 2006, 35(2): 166-170.

[3] 欧阳刘彬, 孙衍华, 刘继凤. GridMol 系统中蛋白质可视化与建模的性能优化[J]. 计算机工程, 2009, 35(20): 242-245.

[4] 王静秋, 王国忠. 基于 Java 3D 的交互式三维动画的研究[J]. 微机发展, 2011, 21(9): 148-152.

[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): 24-26.

[7] 孙茂元. Java 3D 在教育培训类虚拟现实中应用的研究[J]. 现代教育技术, 2010, 20(11): 134-137.

[8] 胡小利, 王炳, 朱惠民. 射弹引起的鱼雷自导系统毁伤仿真[J]. 兵工自动化, 2015, 34(5): 11-13.

[9] 王非, 赵强, 唐定勇. Java3D 文档信息的可视化[J]. 兵工自动化, 2006, 25(4): 91-92.

[10] 陈锋, 李尚会, 沈卓, 等. 基于虚拟仪器的机械压力机制动性能检测系统研究[J]. 机电工程, 2016, 33(12): 1448-1452.