基于 Java3D 的工业机器人建模与远程控制①

Modeling And TeleControl Of Industrial Robot Base On Java3D

文静华 （贵阳市贵州财经学院信息学院 550004、贵州大学计算机软件与理论研究所 550025）
李 祥 （贵阳市贵州大学计算机软件与理论研究所 5500252）

摘要：美国 SUN 公司推出的 Java3D 使 Java 具备了开发三维视觉的小应用程序和应用程序的功能，这些程序具有与客户通过网络交互的能力，使得 Java 在视觉、互动性上产生了一次飞跃。本文提出用 Maya，3Dmax，AutoCad 等熟知的工具可视化地建模，用 Vrml97 格式输出，用 Sun 公司的 Vrml97 Loader 导入，再用 Java 的 Socket 进行远程控制的方案，进行为工业机器人等产品设计阶段的外形设计、动作合理性的检测以及远程控制接口的设计等方面提供一条节省投资的可行途径。

关键词：Java3D 工业机器人 远程控制

1 引言

机器人技术是综合了计算机、控制论、机构学、信息和传感技术、人工智能、仿生学等多学科而形成的高新技术，是当代研究十分活跃，应用日益广泛的领域。Java3D 是美国 Sun 公司为了弥补 Java 语言在多媒体及图形编程方面的不足而推出的一系列标准扩展 API，Java3D 的推出使 Java 开发人员具备了开发三维视觉的小应用程序和应用程序，这些程序还具有与客户交互的能力。这使得 Java 在视觉上，互动性上产生了一次飞跃。基于 Java3D 的这种特性，本文在利用 Java3D 便利的、面向对象的三维图形程序设计来实现对工业机器人的建模与远程控制；本文讨论说明，Java3D 为工业机器人产品设计阶段的外形设计，动作合理性的检测，以及远程控制接口的设计等方面给出了一条节省投资的途径。

2 工业机器人技术简介

现代机器人的历史是从 1959 年美国英格伯格和德沃尔制造出世界上第一台工业机器人开始的。机器人的发展大致经历了三个成长阶段。第一代为简单个体机器人，第二代为群体劳动机器人，第三代为类似人类的智能机器人。

机器人的外形是多种多样的，但有一个共同点：服从人类指令，代替人力自动工作。当然，由于机器人的智能仍然有限，所以人类有时需要对其进行必要的控制，例如探测机器人。美国著名科普作家艾萨克·阿西莫夫为机器人提出了三条原则，即"机器人三定律"：

第一定律——机器人不得伤害人，或任人受到伤害而无所作为；

第二定律——机器人应服从人的一切命令，但命令与第一定律相抵触时例外；

第三定律——机器人必须保护自身的安全，但不得与第一、第二定律相抵触。

这些"定律"构成了支配机器人行为的道德标准，机器人必须按人的指令行事，为人类生产和生活服务。

工业机器人是机器人家族中最重要的成员，它广泛地被应用于各种自动化的生产线上。工业机器人由操作机（机械本体）、控制器、伺服驱动系统和检测传感装置构成。操作机基本上分为两大类：一类是操作本体机构，它类似人的手臂和手腕，配上各种手爪与末端操作器后可进行各种抓取动作和操作作业，工业机器人主要采用这种结构。另一类为移动型本体结构，主要目的是实现移动功能，主要有轮式、履带式、足腿式结构以及蛇行、蠕动、变形运动等机构。壁面爬行、水下推动等机构也可归于这一类。控制器通常是由几级计算机系统构成，通过对各个传感器发回的信息进行计算后再发出相应的指令。伺服驱动系统由驱动单元和伺服驱动控制器两个部分组成，伺服驱动控制器负责控制驱动单元带动操作机朝减少偏差的方向动作。检测传感装置的作用是将各种外界环境参数发回控制器，例如温度，气压，受力等，这使得机器人能够与周围环境保持密切联系，对外界的各种变化做出及时的应变。

3 JAVA3D 技术

在 Java 语言推出并大获成功后，Sun 公司开始考虑如何建立一个真正完整强大的 Java 平台。其中达成的一个共识就是要强化 Java1.0 内核平台中功能有限的图形 API。Sun 公司在 Java1.1 将内核增强了不少，但始终缺了一点什么。好在 Sun 公司与其合作伙伴很快开发出了 Java

① 贵州省教育厅自然科学基金项目（黔教科 2002328）

Media 和 Communication API，弥补了 Java 在多媒体编程方面的缺憾。其中两个最大的弱项，2D 和 3D 图形部分，分别由 Java 2D，3D API 填补了空白。不过，Java2D API 从 Java1.2 起是作为内核平台的 API 出现的，而 Java3D 则是在 Java1.2 推出后的稍晚些时候作为扩展 API 出现的。

Java3D 的推出使 Java 开发人员具备了开发三维视觉的小应用程序和应用程序，这些程序还具有与客户交互的能力。这使得 Java 在视觉上，互动性上产生了一次飞跃。

Java3D 的 API 属于低阶的 API，这些低阶 API 为在场景图结构中建立 3D 几何体提供了一种规范的，独立于底层硬件的方法，并提供了场景预编译等优化技术。这些 API 按作用可分为几个大的模块：基本场景图和形体，形体的分组和几何变换，3D 图形文件的导入，光照和材质等等。

4 利用 Java3D 实现工业机器人计算机模拟的建模

在建模之前，需要进行总体外形的设计，绘制出草图，确定各个构件的外形尺寸以及各个关节转动轴之间的距离。这一点非常重要，因为这决定了在虚拟空间中各个构件的相对运动是否协调，合理。这一步完成后就可以进行部件 3D 模型的建立。

在 Java3D 中要构造一个复杂形体（Shape）大体上有两种办法。一种是利用 Java3D 中自有的 API 产生基本形体，然后再将这些基本形体组合。或者计算出这个复杂形体的各个顶点，然后利用生成面的 API 来形成这个形体。这种使用 Java3D API 来生成形体的方法，优点是对形体的操作比较方便，可以使形体在运行中产生变化。缺点在于这种建立 3D 模型的方法不直观，调试形状比较麻烦。另一种方法就是利用 Sun 公司为 Java3D 开发的 Loader 来载入其他格式存储的 3D 模型，这些模型可以在诸如 Maya，3Dmax，AutoCad 等 3D 生成工具中可视化地生成。这种方法的优点显而易见，那就是建模非常简便，缺点在于在程序运行中这些模型的形状是不能改变的。因为本文是对工业机器人进行模拟，所以一些柔性的附件可以省略，余下的均为刚性部件，不会产生形变，所以采用后一种方法是适宜的。

本文中机器人所有部件的 3D 模型均由 3Dmax 建模，输出为 Vrml97 格式，然后利用 Sun 公司的 Vrml97 Loader 导入虚拟空间中。在建立部件模型的过程中需要注意的主要问题是要把形体的坐标系原点调整到这个部件的转动轴心处，这样在装配这些部件时就能比较容易地将其转动轴心调整至关节转动副轴线处。导入 3D 模型需要用到 com.sun.j3d.loaders.vrml97.VrmlLoader 和 com.sun.j3d.loaders.Scene 两个由 Sun 公司提供的类，其中 VrmlLoader 提供了用于载入图形文件的方法，Scene 对象则用来容纳载入的图形。当 Scene 的对象中赋以图形对象后，可使用 TransformGroup 对象装载该图形对象中的 3D 模型，以便于今后对该 3D 模型的方位调整。整个过程如下：

```
Scene scene=null;
VrmlLoader loader=new VrmlLoader();
TransformGroup module = new TransformGroup
();
module.setCapability(TransformGroup.ALLOW_
TRANSFORM_READ);
module.setCapability(TransformGroup.ALLOW_
TRANSFORM_WRITE);
try
{
  scene = loader.load(fileName); //fileName 为
外部图形文件名，本文中使用的为 vrml 文件 *.wrl。
  module.addChild(scene.getSceneGroup());
}
catch(Exception e)
{
  System.err.println(e);
  System.exit(0);
}
```

当机器人各个部件建模完成并导入场景后，便可以利用 Java3D 中形体变换类 TransformGroup，Transform3D 来组装这些部件。但是这里存在两个问题，那就是如何组装才能使每一个部件自身的运动对其他部件的影响如何才能正确体现出来，并且这些部件能够正确动作？对于第一个问题，我们先举人的手臂为例来说明什么是一个部件对另一个部件的正确影响。当人的上臂活动时，小臂应当随着上臂移动，而小臂活动时，上臂未必动作。而本文模拟的工业机器人的运动基本与此相似。在 Java3D 中这个问题可以利用其构成场景图的特点来解决。具体方法是将动作从属于部件 A 的部件 B 的 transformGroupB 加入到 A 的 transformGroupA 中。

而第二个问题中所谓正确的动作是指各个部件能绕正确的轴心转动。本文中虚拟的机器人没有平移运动，只利用各个关节的转动来实现运动目的。因此这些部件正确的运动，最根本的问题是转动的轴心位置要正确。

在 Java3D 中，物体的方位设置通常是通过 TransformGroup 和 Transform3D 两个类实现的。每个 TransformGroup 对象都有自己的一个坐标系，所有作为子节点加入到这个对象中的形体的位置都以这个坐标系为基准，而这个坐标系同时也代表了这个 TransformGroup 对象在父节点的坐标系中的空间位置和方向。这个 TransformGroup 对象如果要调整在父节点中的方位，只需要利用 Transform3D 对象来调整该对象的坐标系位置即可。调整坐标系位置大体上有两种途径，一是直接设置该调整的变换矩阵，

这种方法的优点是步骤简单快捷，缺点是矩阵的计算比较复杂，不易把握尺度；二是通过 Transform3D 类提供的各种方法来达到目的，这种方法的优点是比较直观，更易于理解，不需要复杂的计算，而缺点在于为了实现比较复杂的运动需要配合多个子 TransformGroup 对象。本文中将采取第二种方法，以便读者能更直观地理解该机器人的运动实现过程。

在组装这些部件的 3D 模型时，每一个形体必须载入到一个 transformGroup 中，由于这一形体上可能有需要运动的子形体，而该子形体运动的基准位置通常是在其父形体上的转轴孔处。因此这里产生了一个问题，一个 transformGroup 对象中需要有两个坐标系，一个用于该级形体在父形体中的定位，另一个作为子形体运动的基准。为了解决这个问题，可以为每一级形体建立一个 container，这个 container 将容纳由外部文件导入的形体，以及坐标系调整到下级形体转轴处的一个 Axis 对象，这样将下一级形体的 container 加入到这个 Axis 对象中，即可使下一级形体转动的轴心处于正确位置。一级形体与其下级形体之间的关系如图 1 所示：

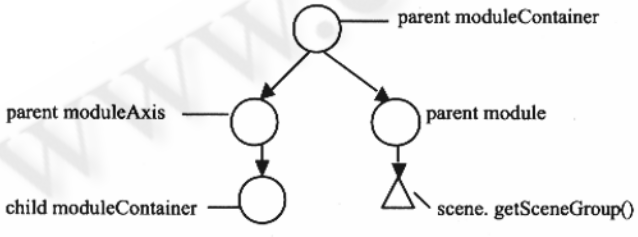

图 1

5　远程控制的实现

机器人的建模部分完成了，接下来的工作是完成机器人的远程控制部分。以本地的机器人对象模拟远程控制的监视窗口，通过 Java Socket 将本地的控制信息同步的传送到 Applet 远程端，那里也实现了一个机器人对象，用来模拟实际的操作机。这里为了简化问题，并没有设置远程端反馈的回路，而是控制端设置为服务器发送消息，远程端设置为客户端接收消息。

该控制系统控制端与远程端的结构分别如图 2（左下图和右下图）所示：

其中控制端的键盘输入接口通过继承 Java3D 的 Behavior 类实现，这个接口将唤醒条件设置为按下按键，然后通过本地实现的 processStimulus 方法分析输入情况，随后根据输入参数调用相应的机器人控制接口使机器人做出响应。

控制端与远程端的机器人控制接口是基本相同的。由于机器钳部分是由两个部件组成，而这两个部件的行为总是对称的，这与其他的部分有显著不同，因此控制接口使用了方法重载来分别处理这两种情况：

```
public void rotArm(float delta); //处理机械钳的运动
```

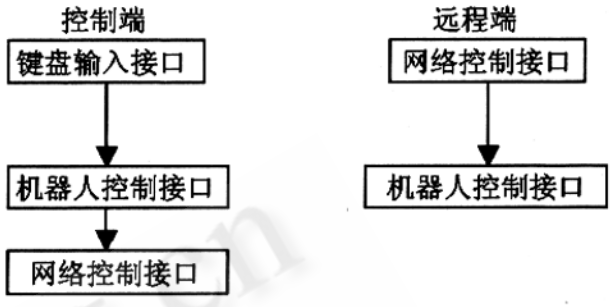

图 2

```
public void rotArm(int code,float delta); //其他部件的运动是相似的，顾客用一个方法来实现
```

在 rotArm 方法中，部件的运动是通过对 Container 做坐标变换实现的：

```
Transform3D t=new Transform3D();
t.rotY(delta); //部件沿哪一根轴转动根据其初始位置来决定
moduleContainer.setTransform(t);
```

在控制端中，对本地对象变换的同时，rotArm 方法调用了网络控制接口。

```
netIo.send(code,delta);
```

控制端的网络接口基于 ServerSocket，在接口中设置有接口状态标志，初始值为接口不可用，当连接建立后，该标志被置为接口可用。消息在发送之前将按照控制协议格式化，远程端在接收到控制消息后，将其解码，然后调用机器人控制接口，实现远程机器人的控制。远程端的网络接口是基于 Socket 的，在程序启动时初始化。控制端与远程端的网络接口均继承了 Thread 多线程类，控制端中子线程用于等待远程端连接，而远程端中则用于持续接收控制端消息。

参考文献

1　Steven Holzner 著，江帆、郑伟、郭春 等译，JAVA2 技术内幕[M]，中国水利水电出版社，2000。
2　Bruce Eckel 著，侯捷译，Java 编程思想[M]，机械工业出版社，2001。
3　Cay S. Horstmann，Gary Cornell 著，李如豹、刚冬梅 等译，Java2 核心技术[M] 卷 I：原理，机械工业出版社，2001。
4　David R. Nadeau，An Introduction to Programming AR and VR Applications with Java 3D[J]，http://www. sdsc. edu/～ nadeau/nadeau _ courses. html2002.5.
5　顾雷、刘鹏、房冰、都志辉 编著，都志辉 主编，刘鹏、陈渝 副主编，Java3D 编程实践[M]，清华大学出版社，2001。