基于 VRML 的 Web3D 环境交互方法①

晏　焱　杨清文　李慧翔　董　飞 (中国人民解放军炮兵学院 安徽 合肥 230031)

摘　要： 基于 VRML 的虚拟现实技术存在交互模式单一、交互控制编程复杂的缺点，根据 VRML 的特点，分析了多事件复杂场景交互和网页与场景交互的方法，提出了运用 API 和网络编程语言实现交互手段多样化的方法，并进行了实际应用，应用表明该方法节约了开发时间、减少了路由层次，提高了运行效率。

关键词： 虚拟现实；VRML；交互性

Interactive Methods of Web3D Pages on VRML

YAN Yan, YANG Qing-Wen, LI Hui-Xiang, DONG Fei (Artillery Academy of PLA, Hefei 230031, China)

Abstract: TVRML has the weakness of haring a single inferactive method and complex control program. To solve this problem, this paper researches interactive methods between Web pages and virtual scenes, and then advances an interactive method by using API and a network programming language. The application of this program indicates the method will condense the development time, cut down route times dprogramming, and improve efficiency.

Keywords: virtual reality; VRML; interactive

1　引言

随着计算机和网络技术的发展，Web3D 在教学、工业、设计制造等许多领域得到了广泛的应用，基于 VRML 的虚拟现实技术，可展示虚拟场景中各部件的位置及其相互之间的关系，并具有文件体积小的特点，易于实现分布式的网络交互环境，但在实现较为复杂的虚拟场景的人机交互时，编程较为复杂并且交互模式单一。本文在 VRML 浏览器插件提供的部分功能函数的基础上，结合运用脚本编程实现了多场景间交互控制和网页对象与场景交互的方法，并通过实例进行了论证。

2　在网页中控制多事件复杂场景的方法

VRML 本身提供的交互是由各节点通过接口域之间的路由(ROUTE TO)实现数据传输，并根据所对应的接口域提取数据改变自身的属性值，从而改变自己在空间中的位置、外型和显示效果[1],其实现原理如图 1 所示。但仅靠 VRML 本身的交互机制是无法构建一个大型交互场景的，特别是多场景之间的交互，因为 VRML 本身不能完成程序设计中的转折、分支、循环等基本特征，因此，大家一般利用 JavaScript 和 Java 语言提供的强大网络编程能力，编写具有远程系统通信和共享能力处理等功能的应用程序。它们对 VRML 的所有支持都通过附加的封装类实现，通过这些类，程序就能够访问 VRML 场景、接受和发送事件、从页面上得到 VRML 对象等，实现对 VRML 场景的完全控制。根据访问方式的不同，可分为：脚本编程接口 SAI (Script Authoring Interface)交互与外部编程接口 EAI(External Authoring Interface)交互[2]。

2.1 脚本编程接口 SAI 交互

SAI 的交互是使用事件机制和路由利用 Script 节点实现 VRML 与 Java(或 JavaScript)的交互。首先，通过 eventIn 将事件传至 Script 节点中的脚本；其次，在 Script 节点中的脚本中调用相应的 Java 类(或 JavaScript)进行处理；最后，通过 eventOut 将结果送回到 VRML 场景以实现动画或交互。VRML 的 Script

① 收稿时间:2010-02-22;收到修改稿时间:2010-05-06

节点是一种控制传感器和内插器的节点，通过它可以定义和改变场景中对象的外观和行为。Script 节点可放置在场景代码的任何位置，用 DEF 命名，在节点的 URL 域中包含一段 JavaScript 程序或链接一个 JavaClass。这种方法编写程序简洁明了，在客户端无需额外的软件支持且兼容性很好，但在实现一些复杂的精确交互时比较繁琐。

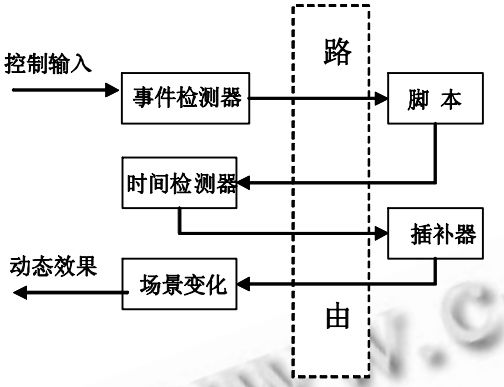


图 1 VRML 简单交互实现原理图

2.2 外部编程接口 EAI 交互

SAI 的交互方式离不开 Script 节点和 Script 类(vrml.node 包)，因此编写的所有 java 类都必须继承自 Script 类，如果想用 Applet 操控 VRML 场景，那么采用继承 Script 类的方法来实现就相当复杂。EAI 则通过 VRML 浏览器的插件接口连接到场景，从而能够访问 VRML 场景的内部节点和事件，实现了 Applet 与 VRML 场景的通信。其特点在于计算功能强大，可实现各种复杂的交互功能，但是它需要借助 Java Applet 实现 VRML 浏览器提供的 Java 接口，由于目前 Web 浏览器对 Java 虚拟机支持的不同，可能导致 EAI 调用不稳定，并且由于这些技术底层依赖于 Java，因此不太适合与 Windows 平台下一些优秀的开发工具(Visual C++、VB、Delphi)相结合进行二次开发。

3 实现网页对象与场景交互以及多场景之间交互控制的方法

支持 VRML 的浏览器很多，但是要实现 VRML 的交互功能还需要安装一个浏览器插件来显示 VRML 文件。现在的 VRML 浏览器插件主要有 Cortona、BS Contact、Cosmo Player、blaxxun 等，不同的插件都针对 VRML 场景的应用开发集成了各自的 API，本文则通过对 BS Contact SDK 中集成的 API 的实验研究，提出了如何实现网页与场景以及同一页面中多场景之间的交互方法。

3.1 网页与场景的交互

通过研究可以看到，VRML 场景的应用中，通过网页元素控制 VRML 场景的例子比较少，BS Contact 插件中的 API 方法 getNodeEventOut（）和 setNodeEventIn（）使问题得到了简化，格式为：\

getNodeEventOut（“NodeName”，“StringName”）

setNodeEventIn（“NodeName”，“StringName”，“Value”）

NodeName 可以是场景中用 DEF 定义的任意节点，StringName 是相应节点的属性名，Value 是属性域值，getNodeEventOut（）方法返回的是指定的域值，setNodeEventIn（）方法是将域值写入对应的属性中，从而达到控制场景的目的。

我们知道，在 html 网页中嵌入 VRML 场景是通过插入标签<embed>或<object>实现的，实际上也是一种 WEB 文档对象，我们可以使用文档对象模型(DOM)中的对象方法对其直接调用，扩展方法 getNodeEventOut（）和 setNodeEventIn（）也是文档对象方法的一种，它们是连接 VRML 场景与网页元素的桥梁，使用这种调用与赋值方法即可实现网页与场景之间的交互，基本语法如下：

ObjectName.getNodeEventOut(“NodeName”，“StringName”)

ObjectName.setNodeEventIn（“NodeName”，“StringName”，“Value”）

2.2 多场景之间的交互

通过对以上方法的试验不难得出：要实现多场景之间的交互，只需为各个场景定义唯一的标识即可，也就是在<embed>或<object>标签中为各个场景定义不同的 ObjectName。定义了不同标识的 VRML 场景对于 DOM 来说就是相互区别的网页元素，使用 getNodeEventOut（）和 setNodeEventIn（）方法就可以对具有不同 ObjectName 的场景进行交互操作了，基本语法如下：

ObjectName1.getNodeEventOut(“NodeName”，“StringName”)

ObjectName1.setNodeEventIn(“NodeName”，“StringName”，“Value”)

…… ……

ObjectNameN.getNodeEventOut(“NodeName”，“StringName”)

ObjectNameN.setNodeEventIn(“NodeName”，“StringName”，“Value”)

经过试验研究表明，getNodeEventOut（）和

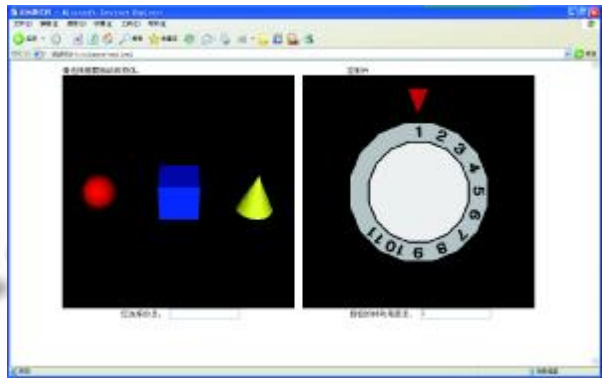
setNodeEventIn（）方法是基于函数入口的，只能在定义的 function（）函数中对其进行调用，而不能直接在 Script 中进行引用。此外，在数据的传递中，getNodeEventOut（）和 setNodeEventIn（）方法中的参数都是 String 类型，在调用时要进行严格的数据匹配。

4　实现示例

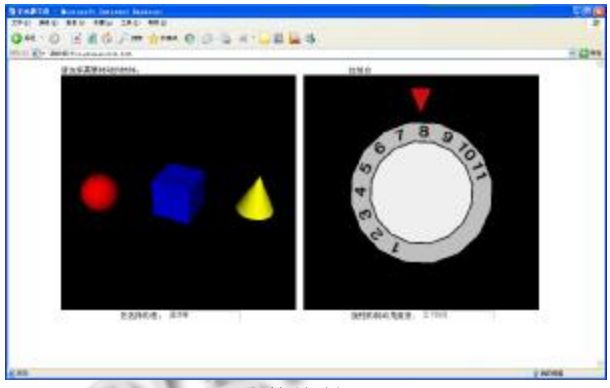
在以上研究与实验的基础上，制作了相应的演示实例，如图 2。在左边的场景中分别定义了三个几何体，右边场景呈现的是一个控制旋钮，点击一个几何体进行选定后就可以通过右边的控制旋钮控制几何体的旋转了，网页下方还可以实时显示被选中物体的名称和旋转角度，弥补了在 VRML 场景中始终不能显示中文信息的不足。

在这个实例中，通过 split（）函数对 getNodeEventOut（）方法获得的字符串数据进行分离，提取各个字符位上的数值加以引用，有效的实现了实时精确的数值显示，通过在场景中加入判断信息节点并结合 JavaScript 中的 setInterval（）函数，实现了对 VRML 场景的监听功能，网页一旦侦听到控制台场景有转动，便由 getNodeEventOut（）方法提取转动参数，再由 setNodeEventIn（）方法直接写入相应几何体节点的旋转属性中，从而实现了右边控制台场景对左边几何体场景的控制。关键代码如下：

```
setInterval("a()",1);   //设定时间函数以实现对函数 a（）的监控，时间间隔为 0.001s
function a(){
var  x;                 //定义变量用于存储模型旋转参数
var parts;              //定义临时变量
x=vrml2.getNodeEventOut("Knob_Cyl","rotation");   //获取旋转参数
parts=x.split(" ");     //将旋转参数进行分离
if(vrml1.getNodeEventOut("panduan","name")=="sphere")
{document.getElementById("textfield").value=" 球体 ";document.vrml1.setNodeEventIn("qiuti","rotation",x);}   //判断选取的对象，实现参数植入
   . . .    . . .
if(vrml1.getNodeEventOut("panduan","name")=="box")
{document.getElementById("textfield").value=" 正方体 ";document.vrml1.setNodeEventIn("fangti","rotation",x);}
document.getElementById("textfield2").value=parts[3];}   //在网页文本域中显示旋转量
```



(a) 旋转控制之前



(b) 旋转控制之后

图 2　多场景交互操作效果图

4　结束语

本文研究了利用浏览器插件中的 API 和脚本编程的方法实现了多场景之间交互控制和网页对象与场景的交互控制，充分发挥了脚本编程灵活和兼容性强的特点，在实时交互控制上提出了新的应用方法，，增加了 VRML 场景的交互样式，为提升 VRML 技术的应用价值进行了有益的尝试。

参考文献

1 阳化冰,刘忠丽,刘忠轩,王庆华.虚拟现实构造语言 VRML.北京:北京航空航天大学出版社, 2000.

2 杨红,龚本.基于 VRML 的虚拟拆装实验技术研究.武汉工程大学学报, 2007,(1):58－60.