

Cortona 组件在虚拟现实系统中的应用

陈轶群 朱群雄

(北京化工大学信息科学与技术学院, 北京 100029)

摘 要: 介绍了如何利用 Cortona 组件技术对 VRML 制作的虚拟现实场景进行动态创建与可靠控制,并引入场景模型数据库,深入探讨了利用场景空间数据进行模型参数修正、数据填充的方法;文章结合笔者研究设计的园区虚拟现实系统平台实例,详细阐述了该技术的核心实现细节。

关键词: 虚拟现实; 模型数据库; VRML; Cortona

中图分类号: TP391

随着网络的蓬勃发展,基于网络的虚拟现实系统(Virtual Reality,简称 VR)日益流行起来。虚拟现实建模语言(Virtual Reality Modeling Language,简称 VRML)已经逐渐成为行业的标准^[1]。VRML 是一种在互联网上构建三维模型的语言,通过嵌入 VRML 浏览器对模型脚本进行解释,可以在普通 Web 浏览器上实现桌面级的虚拟现实系统。作者在开发园区虚拟现实系统平台过程中,采用了 Cortona VRML 浏览器组件,实现了对场景内容的动态创建和自主交互。

1 Cortona 组件的引入

通常,使用 VRML 构建的虚拟场景可以具备“一定层次”上的交互性。在这里“一定层次”是指所有这些场景内模型之间的交互以及今后可能发生的场景与用户之间的交互在模型制作时已经全部设定好了,一旦场景模型设计完成,这些交互行为的模式便不可更改。

解决这类交互问题通常依赖于 EAI(External Authoring Interface,外部程序接口)技术和 SAI(Script Authoring Interface,脚本程序接口)技术^[2],前者需要借助 Java Applet 实现 VRML 浏览器提供的 Java 接口。在利用这些方法对虚拟场景进行控制时,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。

Cortona 组件的出现为处理 VRML 场景交互问题提供了一种新的解决方案。利用该组件提供的丰富的接口,可以方便的对 VRML 浏览和 VRML 构建的场景进行全方位的控制;其 ActiveX 控件的特性使它更容易内嵌于其他开发工具,简化了桌面虚拟现实系统的二次开发过程;在融入模型数据库技术之后,并对元模型进行管理,使场景的空间数据得以更加充分的利用,实现了动态的场景建模与自主式的人机交互。

目前,虚拟园区、虚拟校园等中小型规模的虚拟场景在网络上应用广泛^[3]。通过对虚拟场景模型的设计,利用虚拟现实技术在表现力方面的明显优势,获得了良好的展示效果;这些系统对场景模型的控制有限,交互层次也比较低。为此,笔者进行了园区虚拟现实系统平台的开发和研究。设计并实现了场景实体模型的数据库,由平台负责实体模型与对应空间数据、属性数据之间的绑定,并利用 Cortona 组件对空间数据进行操作,显著提高了系统的交互能力。

2 实体模型数据库的设计与实现

建立实体模型数据库的核心目标是最大限度的实现模型复用。在 VRML 构造的虚拟场景中,模型复用具有重大的意义。

模型复用利于模型的管理。抽取场景中具有相同外观特征的实体模型,进行分类存储,在很大程度上简化模型的修改与使用。例如,在园区设计中,设定了树木模型、路灯模型这样的大分类,在这些分类

收稿日期: 2004-11-21

第一作者: 男,1981 年生,硕士生

E-mail: cheniyiqun@mail.buct.edu.cn

中,添加了 Tree1、Tree2 以及 Lamp1、Lamp2、Lamp3 这样几个基本的元模型(元模型文件的产生可以通过其他 3D 造型软件来完成,如 AutoCAD、3DMAX 等,最终的文件后缀名为 .wrl),这些模型在场景中可能会大量的重复出现,通过建立并不断扩充模型库,可以方便的向场景中加入模型;同时,对模型的修改、删除操作则直接对模型库中相应的元模型进行操作即可。

模型复用能够显著压缩生成的 VRML 文件的大小。Inline 节点是 VRML 语言提供的标准节点之一,它负责将外部的既有模型引入到当前场景中;该节点支持嵌套引用,并常与 Transform 节点配合使用,后者控制模型引入的空间位置、缩放比例及旋转等参数,通过指定元模型的存储路径名,就可以在场景中加载相应的实体模型。模型复用性越高,主 VRML 文件结构性越好,其体积也越小。

实体模型库的基本结构如图 1 所示。

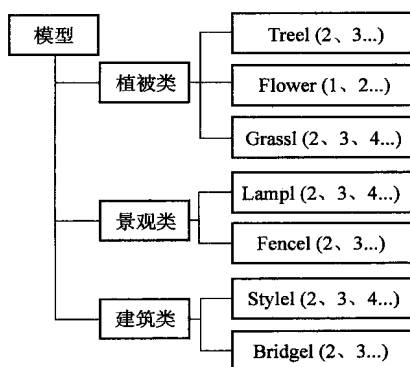


图 1 模型库结构示意图

Fig. 1 Structure of model database

模型库从磁盘存储的角度上看由两部分组成,一是存放 *.wrl 文件的目录存储结构,一是将模型的分类、名称与其存放路径建立关联的数据表结构。园区虚拟现实系统平台中有专门的模型库管理模块负责所有元模型的管理。

3 处理模型参数修正问题

利用模型库机制进行模型复用的优越性十分明显,但在具体实施过程中,元模型自身尺寸与整体场景之间的匹配问题就暴露出来了。由于各个不同元模型的尺寸大小各异,若直接引入,容易造成场景内模型间比例失调。

为统一模型间的比例,需要利用 Cortona 组件中的 BoundBox 方法。在 VRML 中,每一个模型文

件在虚拟空间内表现为若干节点的组合与嵌套, BoundBox 方法可以获得该模型的外围盒,外围盒以其包含的模型几何体在 X 、 Y 、 Z 方向上的坐标上下限构造自身大小,它反映了模型在虚拟空间中所占据的体积。在平台中进行模型搭建时,先为要引入的模型指定一个放置区域(为任意多边形,面积为 S)与在场景中的高度 H ,通过简单的几何运算,计算出 S 的外围矩形面积 S' ,用引入模型的 BoundBox 的体积去逼近在场景中由 S 、 H 定义的一个体积,并将获得的缩放因子 R 传递给接收引入模型的 Translation 节点的 Scale 域,这样就解决了引入模型的比例统一问题,具体过程如图 2 所示。

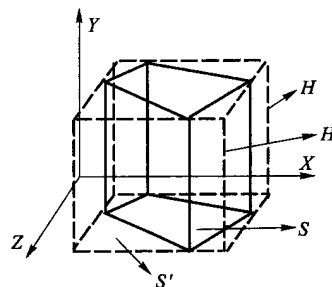


图 2 使用外围盒技术解决比例缩放问题示意图

Fig. 2 Schematic of handle of scaling problem by bound box technique

上图中,实线围成的体积是实际放置区域的体积($V = SH$),虚线围成的体积是以 S 的外围矩形 S' 为底面的长方体($V = S'H$)。用 Cortona 组件获得引入模型 BoundBox 的体积 V_m ,并用 V_m 等比例去逼近 V ,获得缩放因子 R ,再将 R 传递给 Translation 节点的 Scale 域,调节引入模型的比例尺寸,从而修正了模型的比例参数问题。

4 处理场景数据填充问题

在利用建模平台进行场景搭建时,关联的数据库不仅记录下场景内部各模型的属性信息也记录下模型的空间位置信息;道路在平台设计中由一系列的关键点构成,不同道路上的关键点可以重叠以实现交叉路口,通过这种方法可以生成任意复杂的道路网;在将场景内的地物模型空间位置信息与其相邻的道路关键点进行绑定之后,应用合适的路由算法并利用 Cortona 组件的对场景数据的填充能力,就可以实现场景内任意指定位置的视角定位以及任两点(地物)间的自动导航功能。功能结构图如图 3 所示。

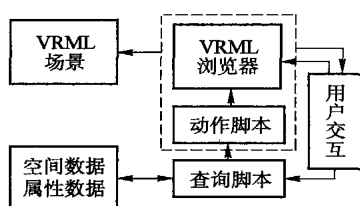


图 3 虚拟园区场景交互示意图

Fig. 3 Schematic of scene interaction

用户可以通过鼠标、键盘直接对 Cortona VRML 浏览器进行控制,当用户选中(选中在虚拟环境中表现可为触摸、碰撞,通过 VRML 内置的传感器节点,可以实现对用户动作的捕捉)场景内的地物时,查询脚本将对关联的数据库进行搜索,首先获得地物的属性数据,并组织成 HTML 页面显示给用户;接着动作脚本调用 Cortona 组件接口,获取指定地物的外围盒中心坐标及尺寸参数,计算并修改场景视角堆栈中栈顶视角的 Position 域值

根据地物的外围盒中心 $bbCenter$ 及尺寸 $bbSize$ 参数计算其匹配的视角坐标,输出是视角坐标向量。 $VecVPP = ToVPPosition(bbCenter, bbSize)$

调用 Cortona 组件改变栈顶视角的 Position 域值
Cortona. Engine. Viewpoint. Fields(" Position "). $x = VecVPP.x$;

Cortona. Engine. Viewpoint. Fields(" Position "). $y = VecVPP.y$;

Cortona. Engine. Viewpoint. Fields(" Position "). $z = VecVPP.z$;

同时还要计算出一个合适的观察角度,改变栈顶视角的 Orientation 域值

根据地物的外围盒中心 $bbCenter$ 及尺寸 $bbSize$ 参数计算匹配的视角坐标,输出含四个参数的向量, x, y, z 指定旋转轴,angle 指定旋转角度。

$VecVPO = SetVPOrientation(bbCenter, bbSize)$

调用 Cortona 组件改变栈顶视角的 Position 域值
Cortona. Engine. Viewpoint. Fields(" Orientation "). $x = VecVPO.x$;

Cortona. Engine. Viewpoint. Fields(" Orientation "). $y = VecVPO.y$;

Cortona. Engine. Viewpoint. Fields(" Orientation "). $z = VecVPO.z$;

Cortona. Engine. Viewpoint. Fields(" Orientation ").
 $angle = VecVPO.angle$;

这样用户就会以一个合适角度观察选中的地物

了。图 4 是用户在用鼠标选中园区场景内小区超市后的展示页面。

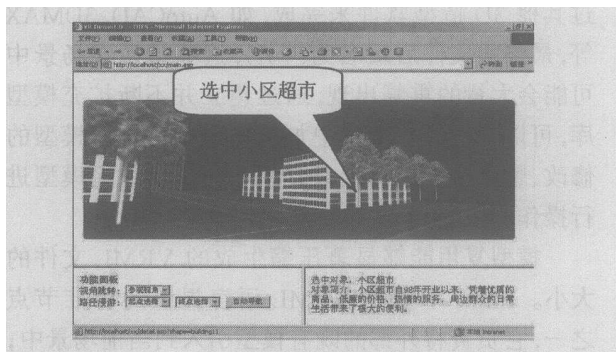


图 4 用户选中小区超市

Fig. 4 Supermarket selected by user in the community

目前在大多数虚拟园区场景中,只可以进行若干条固定路线的自动导航,场景无法实现任意指定两点的自动导航。但结合了数据库技术以后,便可以充分利用模型的空间数据信息^[4],再加以 Cortona 组件提供的接口去填充场景数据,便可以实现自主式的导航功能。

当用户在控制页面中选定导航的起始位置和目标位置以后,查询脚本将这两点传递给自定义的路由组件,由路由组件对数据库中的道路节点进行路由选择,并将路由结果返回给查询脚本,路由结果中包含了途经的所有关键点空间坐标以及在每个关键点时视角的旋转状态,这些数据在完成格式处理之后会继续传递给动作脚本。

首先,动作脚本会通过分析路由关键节点的数目来定义适当的整个路由动画完成的时间

动作脚本接收到的是两个向量数组,PIArray 数组中每个元素包含三个分量,对应关键点的 x, y, z 坐标;OIArray 数组中每个元素包含四个分量,分别对应应在关键点位置处视角的旋转轴和旋转位置 $x, y, z, angle$;获得关键点的数目

$KeyPointCount = UBound(PIArray)$

设置路由动画时间,其中 Path Timer 为 TimeSensor 类型的 VRML 节点

Cortona. Engine. Nodes(" PathTimer "). Fields(" CycleInterval ") = $KeyPointCount * 1.5$

接着,动作脚本将分别对场景中的两个内插器 PI 和 OI 进行赋值。为了实现自动导航,在 VRML 脚本中设置了 PositionInterpolator 内插器和 OrientationInterpolator 内插器,这两个内插器在 TimeSensor 节点的协调控制下,与 ViewPoint 节点进行

绑定,实现在场景内的自动导航。

根据路径关键节点数,循环插值,填充数据

For ($i = 0; i \leftarrow \text{KeyPointCount}; i++$)

{ 先设定两个内插器的 Key 域

Cortona. Engine. Nodes

("PI"). Fields ("Key"). Add ($i / \text{KeyPointCount}$);

Cortona. Engine. Nodes

("OI"). Fields ("Key"). Add ($i / \text{KeyPointCount}$);

再设定两个内插器的 KeyValue 域

Cortona. Engine. Nodes

("PI"). Fields ("KeyValue"). Add (PIArray [i]);

Cortona. Engine. Nodes

("OI"). Fields ("KeyValue"). Add (OIArray [i]);

同时,场景还将自动创建一条深灰色导航路标,图 5 为自动漫游的起点。

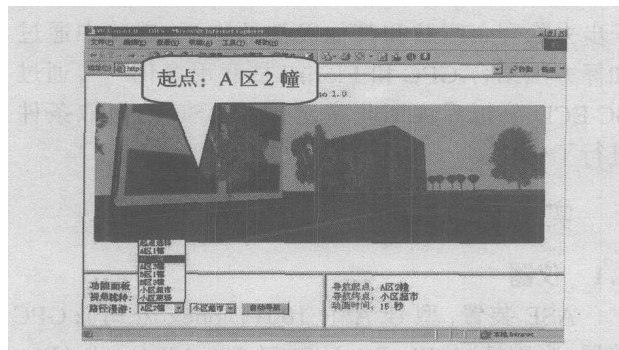


图 5 场景自动漫游起点 A 区 2 幢

Fig. 5 Routing start point: 2nd building of A block

图 6 显示的是路径漫游中的情形,深灰色路标指向处就是漫游的终点小区超市。



图 6 场景自动漫游终点小区超市

Fig. 6 Routing end point: supermarket in the community

5 结束语

在引入 Cortona 组件并结合模型数据库技术以后,利用 Cortona 组件提供的丰富接口,实现了场景内视角跳转和动态漫游的创建,较充分的利用了场景模型的空间数据,为虚拟现实技术的实现提供了一种便捷、高效的新方法。

参 考 文 献

- [1] 王发坤,崔汉国. VRML 在 Web 仿真中的应用[J]. 计算机工程, 2002, 28(8): 50 - 53
- [2] 王全科,刘岳. VRML 在三维动态交互地图可视化中的实践[J]. 中国图象图形学报, 2001, 6(3): 15 - 19
- [3] 王德新,魏东,黄有群. 在 VRML 文件中实现对数据库信息的访问[J]. 沈阳工业大学学报, 2002, 24(5): 417 - 420
- [4] 万剑华,潘正风,李清泉. 基于 VRML 的虚拟城市的建立[J]. 测绘通报, 2002(5): 17 - 19

Application of Cortona ActiveX to development of a virtual reality system

CHEN Yi-qun ZHU Qun-xiong

(College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China)

Abstract: Application of the Cortona ActiveX to a virtual reality system developed by VRML with dynamic generation and reliable control of the virtual scene was introduced as well as the design of a model database for fully utilizing the special data and methods for the model revision and the data filling were discussed. Based on the development of a community virtual reality system as an example, the key technique for development was given in more detail.

Key words: virtual reality; model database; VRML; Cortona

(责任编辑 刘同帅)