

基于 3DMouse 的虚拟仿真系统人机交互技术

王海东, 朱文华, 苏玉鹏

(上海大学 CIMS 与机器人中心, 上海 200072)

摘要: 为了提高虚拟仿真系统的人机交互性, 增强虚拟现实的沉浸感, 并结合对虚拟装备仿真系统人机交互性能的需求分析, 提出了一种基于 Open Inventor 技术的三维鼠标视点控制方法。利用三维鼠标的操作模式与工作原理, 建立了一种新的仿真系统人机交互模型, 从而解决了三维鼠标作为人机交互设备与虚拟仿真系统集成的问题, 实现了操作者操纵三位鼠标在虚拟场景中的漫游和装备三维对象模型的动态控制。三维鼠标浏览路径数据的保存和导入功能可实现反复多角度的观察虚拟装备的装配和工艺过程, 使装备的操作和维护培训效果显著, 该技术可应用在多种虚拟装备仿真系统中。

关键词: 人机交互; 虚拟仿真; 三维鼠标; 视点控制

中图分类号: TP391.9

文献标识码: A

文章编号: 1673-629X(2009)10-0018-04

The Human - Computer Interaction Technology of Virtual Simulation System Based on 3DMouse

WANG Hai-dong, ZHU Wen-hua, SU Yu-peng

(CIMS & Robot Center of Shanghai University, Shanghai 200072, China)

Abstract: For improving human - computer interaction of virtual simulation system and enhancing immersion feeling of virtual reality, put forward a kind of 3Dmouse view point control method based on Open Inventor technology through the requirement analysis on human - computer interaction of virtual equipment simulation system, and construct a new human - computer interaction model of simulation system. Achieve roaming in virtual scene and dynamical control to the 3D object model with the operation by operator, through solving the problem of integration between human - computer interaction equipment and virtual simulation system. The function of saving and importing of 3Dmouse browsing path data realize viewing equipment's assembly and work process from many angles repeatedly, which make the efficiency of operation and maintenance more striking, this technology can be used in many virtual equipment simulation systems.

Key words: human - computer interaction; virtual simulation; three - dimensional mouse; view point control

0 引言

三维鼠标一直是虚拟现实技术中人机交互技术研究的热点。传统的三维鼠标, 以 Logitech 三维鼠标为例, 其主要由发射器、接收器和控制单元组成^[1]。由于结构复杂, 控制局限性较高, 跟踪的精度也并不能达到实际的需求。近年来, 随着虚拟现实技术、人机交互技术以及其它计算机技术的发展, 这类跟踪设备也有了很大的发展。目前已经出现了许多功能完善且效果逼真的交互输入输出设备, 这些设备使人机交互变得更加自然、和谐。

为了提高虚拟装备三维场景和模型的跟踪与控制精度, 达到结构简单、易操作易控制的目的^[2], 采用了 3D Connexion 的 SpaceBall5000 型号三维鼠标及 3DWare SDK 开发包, 结合 Open Inventor 视点对象控制技术, 将三维鼠标假定为三维场景中特殊的控制节点, 对三维场景树的视点节点进行重构, 在 Windows 操作系统环境下, 以 Visual Studio 2003.NET 的 MFC 框架为基础, 应用 VC++ 语言进行程序开发, 提出一种基于 Open Inventor 的视点控制方法, 通过此方法构建了虚拟装备仿真系统的人机交互模块, 实现了仿真系统三维场景和对象简便、高精度的动态控制, 在实际应用中达到了良好的效果。

1 SpaceBall5000 三维鼠标操作模式及原理

如图 1 所示, 系统所采用的 SpaceBall5000 型三维鼠标提供了良好的六自由度的交互控制, 能够监测到

收稿日期: 2009-02-18; 修回日期: 2009-05-17

基金项目: 上海市科技重大项目基金资助项目(05DZ11001); 上海市重点学科建设资助项目(Y0102)

作者简介: 王海东(1984-), 男, 辽宁人, 硕士研究生, 研究方向为虚拟制造; 朱文华, 博士, 教授, 研究方向为数字化设计制造技术、虚拟仿真等。

手指施加其上最轻微的力,并将其分解成 X, Y, Z 方向的平移与转动^[3]。该三维鼠标的平移方向和旋转轴如图 1 所示(展示的三维鼠标为 SpaceBall3003)。

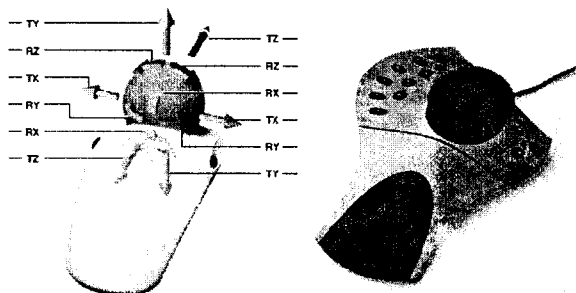


图 1 SpaceBall5000

1.1 三维鼠标的两种基本操作模式

(1) 对象模式:将手中的三维鼠标球想象成要控制的三维模型进行操作,通过将鼠标球向前后左右上下方向的推拉,实现三维对象在相同方向的平移运动,同样地,通过将鼠标球向任意方向的旋钮,实现三维对象在相同方向的转动^[4]。

(2) 人眼模式或摄影机模式:将手中的三维鼠标球想象成人的头部或者是一部摄影机,将鼠标球向前后左右上下方向推拉,三维场景会向其运动的相反方向移动,对鼠标球的转动操作也是如此,实现在三维场景中的漫游。

1.2 三维鼠标超感应器技术基本原理

三维鼠标球作为一个中空的外球(鼠标球表面)恰当地“悬浮”在六根叶片弹簧上,弹簧的一个球形顶端固定在鼠标外球的相应夹槽中。一旦鼠标外球受到推拉(或者/和)扭动,叶片弹簧将发生偏移以抵消外力,将施加在鼠标球上的平移力和扭转力分解成六个力。这时发光二极管 LED 监测器组利用光遮传感技术就会测量出每个叶片弹簧的偏移量,从而检测鼠标球受力后的位移^[5]。

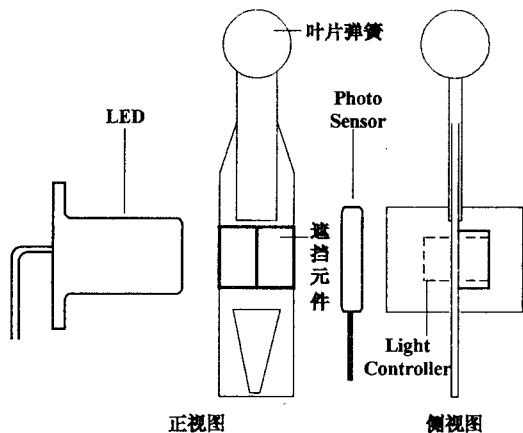


图 2 光学传感器简图

如图 2 所示,在光学传感器中最左边的元件是发

光二极管 LED,叶片弹簧右侧的部件是图像传感器 Photo Sensor,其上的虚线区域遮挡了来自 LED 的光。光学传感器设置在每根叶片弹簧的底端,每一根叶片弹簧铸有一个光学遮挡元件,它遮挡了部分光的照射。在未受力状态下,一半的光被遮挡,当弹簧朝某一方向的偏离使得更多的光通过,而另一方向则更多的光被遮挡。光的多少被用来计算弹簧的偏离量,将每根轴上得到的一对力的值相加就得到了力矢量,相减就得到了扭转矢量。

2 基于 Open Inventor 的三维鼠标视点控制方法

Open Inventor 是用于实现交互式 3D 图形应用的全方位、面向对象的框架体系,拥有功能强大的基于 Win32 的 IVF 类库。

虚拟装备场景中的全局视点节点定义了在场景区排在它后面节点的观察方式。视点控制方法的总体程序设计思想是,通过调用 Open Inventor 中基本视点对象 SoCamera 的子类 SoPerspectiveCamera 视点控制对象来实现虚拟场景的重绘^[6]。将 SoPerspectiveCamera 对象节点放置在虚拟场景树 scene graph 的左上部,利用 Perspective Camera 的投射模拟人眼观测的真实情况,使虚拟场景中需要显示的对象节点排在它之后,这样整个虚拟场景的显示,例如对象的位置、方向、屏幕高宽比、裁剪平面的位置、高度角都可以通过修改 SoPerspectiveCamera 节点的相应域值来实现统一灵活的操作控制。

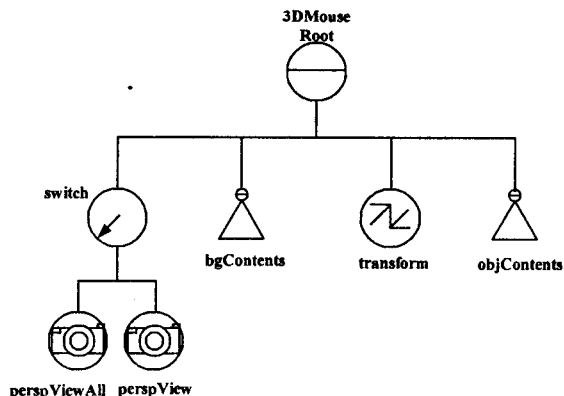


图 3 视点控制模型

但是经过试验,要最终实现虚拟场景视点浏览控制功能,必须在此基础上(如图 3 所示)抽象出两个 SoPerspectiveCamera 对象 perspViewAll 和 perspView 作为子节点添加到 Switch 组节点下,在程序实现过程中通过 Switch 节点控制子节点属性的切换^[7]。由于三维控制只针对装备体,而非整个三维场景,因此将坐标

变换 SoTransform 节点对象 transform 插到 bgContents 背景对象与 objContents 装备对象之间,实现对象控制转换。

为了将三维鼠标作为整个虚拟场景的控制引擎,必须在程序的设计过程中将其作为整个三维场景的根节点,控制节点的属性域值。

3 视点控制方法实现系统人机交互模块

通过以上视点控制方法,结合三维鼠标的 SDK,程序中定义三维鼠标类(C3DMouse)的主要成员如下^[8]:

```
SoPerspectiveCamera * perspViewAll; //视点对象
SoPerspectiveCamera * perspView; //视点对象
SoSwitch * switch; //switch 节点对象
SoTransform * Transform; //坐标变换节点对象
void addObjectChild(SoSeparator * root, SoSeparator * object =
NULL); //添加控制对象
void addEnvironmentChild(SoSeparator * root, SoSeparator *
environment = NULL); //添加背景对象
void ViewToOrFromObject(); //三维鼠标控制切换,视点与
物体控制相互切换
void Reset(); //C3DMouse 对象数据重置
SiHdl m_ DevHdl; //三维鼠标设备句柄
LONG
OnSpaceball(WPARAM wParam, LPARAM lParam, int WM_
SPACEBALL); //三维鼠标消息接收函数 int SbInit(CWnd *
pWnd); //三维鼠标初始化
void SbButtonReleaseEvent(int buttonnumber); //释放三维鼠
标按键处理函数
void SbButtonPressEvent(int buttonnumber); //三维鼠标按
键消息处理函数
void SbZeroEvent(); //重置三位鼠标数据
void SbMotionEvent(SiSpwEvent * pEvent); //三维鼠标空
间球运动消息处理函数
void C3DMouse:: SaveBrowsePathToFile(char * filename) //
将浏览路径中的视点路径存入文件
void C3DMouse:: AddBrowsePathFromFile(char * file-
name) //将视点路径文件导入浏览路径中
```

3.1 三维鼠标及虚拟场景的初始化

三维鼠标以 MFC 的消息响应机制作为通信方式,即当操作事件被触发时,Windows 接收到该事件的消息,并触发对应的消息响应函数。要实现该过程必须首先向 Windows 注册三维鼠标的消息,然后针对该消息创建对应的消息响应函数,这样系统就能自动识别三维鼠标的消息,并调用响应函数。三维鼠标的操作事件主要包括控制球操作和按钮操作,前者用于控制视点或虚拟装备的六自由度运动,后者用于设定运动

模式、调整运动精度和启动控制界面等。注册三维鼠标的程序如下^[8]:

```
MagellanControlInfoDLL. MagellanMotionEvent = Register-
WindowMessage("MotionEvent"); //处理控制球操作事件
MagellanControlInfoDLL. MagellanButtonPressEvent = Register-
WindowMessage("ButtonPressEvent"); //处理按钮按下事件
MagellanControlInfoDLL. MagellanButtonReleaseEvent = Reg-
isterWindowMessage("ButtonReleaseEvent"); //处理按钮抬起事
件
```

装备虚拟场景初始化后,系统就开始检索三维鼠标,并对三维鼠标类进行初始化,创建节点 SoPerspective 和 SoTransform 类对象,建立视点节点和装备本体的位姿变换节点之间的联系。

3.2 视点控制实现过程

首先,判断事件类型,如果是控制球操作事件,则从外部接口中获取三维鼠标输出的方位数据,通过 C3DMouse 类对数据进行处理,根据预设定的运动限制模式和精度控制比例因子修正该方位数据,如果仿真系统的坐标系与三维鼠标的坐标系不同,则还要修正坐标方位值,得到新的方位数据^[9]。然后,将处理后的数据作为参数传递给对 Open Inventor 节点域值控制的响应函数,改变虚拟场景和对象的运动状态和方位,达到控制视点或装备运动状态的目的。如图 4 所示,每当操作者改变三维鼠标的当前状态时,消息循环就会触发对视点节点或装备本体位姿变换节点的相应操作,从而实现场景视点或装备位姿的变化。

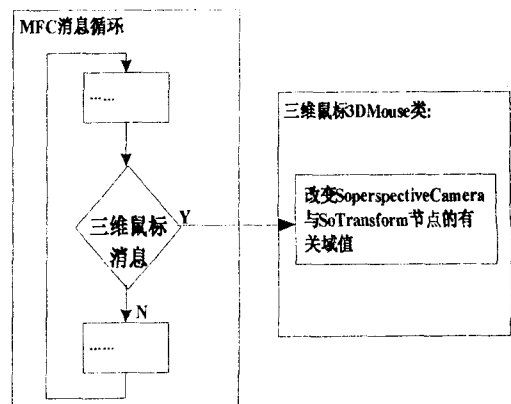


图 4 虚拟场景中视点、装备的控制视图

以控制球控制视点对象实现虚拟场景漫游过程为例,解释三维鼠标消息响应函数的实现过程:

(1) 三维鼠标初始化:

创建 SoPerspectiveCame, SoSwitch, SoTransform 视点控制对象,建立视点控制模型。

```
mouse -> addEnvironmentChild(root, background_ root);
```

```
mouse -> addObjectChild( root, Scene_ root);
```

mouse 是三维鼠标类 C3DMouse 的对象,addEnvi-

ronmentChild 和 addObjectChild 两个方法将 Open Inventor 虚拟环境中的背景节点和三维实体对象节点添加到三维鼠标的根节点下,将三维鼠标作为一个独立的节点扩展了 Open Inventor 的节点库,搭建了上述视点控制模型节点树结构。

(2)注册三维鼠标控制球消息:

```
int WM_SPACEBALL = RegisterWindowMessage ("SpaceWareMessage00");
```

当控制球事件触发时,系统自动自获消息,从而调用相应的消息响应函数。

(3)进入装备训练界面后,可以操作控制球在场景中漫游,程序实现如下:

1)控制球响应函数判断事件的消息类型,调用控制函数:

```
LONGC3DMouse::OnSpaceball (WPARAM wParam, LPARAM lParam, int WM_SPACEBALL)
{
    if (SiGetEvent (m_DevHdl, 0, &EData, &Event) == SI_IS_EVENT)
    {
        switch (Event.type)
        {
            case SI_MOTION_EVENT:
                SbMotionEvent (&Event); /* 控制球事件调用函数 */
                break;
            ....
        }
    }
    return (TRUE);
}
```

控制球响应函数判断三维鼠标注册的所有消息,根据标识,调用具体方法获取虚拟场景中的坐标和角度数据,通过具体算法得出 Open Inventor 控制节点的域值,实现虚拟场景或对象的位置和状态变换。

2)控制球响应函数调用 SbMotionEvent(&Event) 函数,根据消息事件的地址参数,结合预定义的判断标识进一步调用实现节点域值变换的视点控制函数:

```
void C3DMouse::MoveViewPoint(int direction, int range)
{
    switch(direction)
    {
        case 1:
            SVec3f initialPos;
            perspViewAll->orientation.getValue(initialPos,angle);
            initialPos = perspViewAll->position.getValue();
            float x, y, z,x_add,z_add;
            initialPos.getValue(x,y,z);
```

```
x_add=sinf(M_PI/2+angle)*(angle/20);z_add=cosf(M_PI/2+angle)*(angle/20);
perspViewAll->position.setValue(x+x_add,y,z+z_add);
break;
....
}
```

direction——视点 x, y, z 正负方向平移旋转以及复位标识,range——运动幅度。虚拟环境的场景树节点中,视点控制对象节点控制着其后所有的场景节点的显示,利用 Open Inventor 视点控制对象 perspViewAll 获取当前视点坐标和角度,通过计算坐标的增量值控制整个场景节点的域值,实现虚拟场景和装备对象的状态和位置变换,达到视点漫游的目的。其余消息的响应函数实现过程与上述过程类似,具体算法在此不再赘述。

4 结束语

开发过程中,对 Open Inventor 的节点库进行了扩展,实现了三维鼠标作为一个独立的节点控制视点及其他节点域值的功能。文中涉及的程序,在 PC 机 Windows XP 下,用 Visual Studio 2003. Net 调试通过。将三维鼠标人机交换技术应用在 PECVD(等离子体增强化学气相沉积)仿真系统装配演示和维护培训中。如图 5 所示,通过视点浏览功能操作三维鼠标控制球在虚拟场景中漫游达到从多角度进行观察的目的。在程序的开发过程中还实现了视点浏览路径的保存和导入功能,可以在装备的维护培训中通过重复演示达到良好的效果。

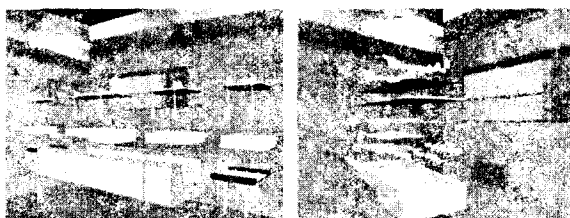


图 5 视点漫游演示

实践证明该系统具有直观的可视效果、增强了虚拟环境的沉浸感,并能很好的适应装备安全维护和操作培训的目的。

参考文献:

- [1] 詹荣开,王宏伟,任敏. Logitech 三维鼠标及其在三维场景漫游中的应用[J]. 计算机工程,2001,27(1):187-189.
- [2] 杨旻,阮秋琦. 基于 MultiGen 的虚拟场景真实感研究

(下转第 25 页)

(1) 当 MN 进入一个新的 MAP-AAA_c 域时,认证切换处理包括以下几个步骤:根据路由器宣告消息,配置新 LCoA 和 RCoA。

(2) AAA_h 利用根据网络地址标识符 (NAI) 查找到的用户信息生成一组认证矢量 AVs, 同时将 BU_{HA} 消息转发给 HA, HA 对新 RCoA 进行绑定更新。

(3) AAA_h 将 AVs 和 BA_{HA} 应答给 MAP-AAA_c; MAP-AAA_c 存储这组 AVs。

(4) MAP-AAA_c 取出一个 AV, 将其 RAND 和 AUTH 发送给 MN, 代替 AAA_h 进行认证。

(5) MN 利用 AUTH 对网络认证成功后, 根据 RAND 和它与 AAA_h 共享的 SA 计算出 RES, 发送给 MAP-AAA_c, 同时计算出 CK 和 IK (实现后续通信数据传输安全性保障的会话密钥)。

(6) MAP-AAA_c 比较 AV 中的 XRES 和从 MN 收到的 RES。相同, 则 MN 通过认证, 向其应答认证成功消息, 并捎带 LCoA 和 RCoA 的绑定确认消息; 不同, 则 MN 认证失败, 向其应答认证失败消息。

这样就完成了一次认证和移动注册, 同时实现了 CK 和 HK 在 MAP-AAA_c 和 MN 的发放。

当 MN 在 MAP-AAA_c 域内的不同 AR 间移动时, MAP-AAA_c 实现认证和注册本地化, 无需与家乡域进行交互, 具体步骤如下:

1) MN 首先获得一新的 LCoA。

2) MN 向 MAP-AAA_c 发送注册请求信息。

3) MAP-AAA_c 执行移动锚点功能, 对 MN 的新 LCoA 进行绑定更新。

4) MAP-AAA_c 执行认证锚点功能, 取出保存的 AVs 中的一个, 完成认证及会话密钥发放。

5 结束语

通过分析现有 MIP-AAA 方案在切换时延方面存在的不足, 提出一种快速认证注册方案。新方案通过减少认证过程中外地域与家乡域交换的消息数量来

缩短切换时间, 特别是在域内切换情况下, 外地域可以直接认证用户身份, 不再需要家乡域参与认证过程, 极大地提高了切换效率。新方案为实现 AAA 环境下移动 IP 的快速、安全、扩展性好且低成本的认证注册提供了一种思路。随着社会的不断发展, 信息化程度的不断提高, 互联网和移动网络的迅猛发展, 基于移动 IPv6 的实时应用的不断引入, 节点的移动速度不断变快, 而如何提高节点切换和接入认证的效率问题是非常有意义的。

文中利用层次化基于身份签名方案的特性, 将层次化移动切换过程和认证过程进行了有机整合, 减少了访问网络和家乡网络之间的消息交互, 减少了延迟和开销, 提高了认证切换性能。

参考文献:

- [1] 刘淑芝, 吴海涛. IPv6 之后的网络安全问题分析[J]. 计算机技术与发展, 2006, 16(8): 243-245.
- [2] 王 政, 陈 萍. 宽带用户接入认证方式浅析[J]. 山东通信技术, 2002, 22(3): 31-33.
- [3] 林嘉燕, 俞鹤伟. 移动 IPv6 切换技术[J]. 计算机技术与发展, 2008, 18(10): 158-161.
- [4] 陈 蕾, 杨 鹏, 何剑锋. 基于 HMIPv6 的域间无缝切换方案[J]. 计算机工程与设计, 2008, 29(15): 3885-3888.
- [5] Gergiades M, Akhtar N. AAA context transfer for seamless and secure multimedia services over All-IP infrastructures [C]//International Conference on Telecommunications (ICT). Polynesia, France: [s. n.], 2003: 597-603.
- [6] 田 野, 张玉军, 张瀚文, 等. 移动 IPv6 基于身份的层次化接入认证机制[J]. 计算机学报, 2007, 30(6): 905-915.
- [7] Eronen P, Hiller T, Zorn G. Diameter extensible authentication protocol (EAP) application[S]. IETF RFC 4072, 2005.
- [8] 田 野, 张玉军, 刘 莹. 移动 IPv6 网络基于身份的快速认证方法[J]. 软件学报, 2006, 17(9): 1980-1988.
- [9] Soliman H, Castelluccia C, Malki K E, et al. Hierarchical Mobile IPv6 mobility management (HMIPv6)[S]. IETF RFC 4140, 2005.

(上接第 21 页)

- [J]. 计算机技术与发展, 2008, 18(9): 177-179.
- [3] 曹逸峰, 陈一民, 顾文望. 虚拟机床可视化设计相关技术的研究与应用[J]. 计算机技术与应用进展, 2004, 16(8): 520-523.
- [4] 张小超, 王精业. 虚拟场景漫游系统的体系结构分析[J]. 系统仿真学报, 2005, 17(4): 917-919.
- [5] Elsadek H, Eleeb H, Horikoshi J. New technique for investigating three dimensional 3D Mouse[J]. Systems, Man, and Cybernetics, 1998, 5(5): 4344-4347.
- [6] 张向波, 邢朝伟. 人机交互中的场景开发[J]. 微机发展(现

更名: 计算机技术与发展), 2003, 13(12): 98-101.

- [7] 刘 良, 黄路伟. 基于 Open GL Performer 的视图优化研究[J]. 计算机技术与发展, 2007, 17(8): 77-80.
- [8] 3DxWare SDK 1.0 for Windows 98/ME and Windows NT (4.0/2000/XP32)[M]. [s. l.]: [s. n.], 2000.
- [9] Ishii Y, Osaki K, Watanabe T. Evaluation of Embodied Avatar Manipulation Based on Talker's Hand Motion by Using 3D Trackball[C]//The 17th IEEE International Symposium on Robot and Human Interactive Communication. Technische Universität München, Munich, Germany: [s. n.], 2008: 653-658.