

# 基于多相机多视图漫游的三维交互技术的研究及应用

张桂花 段雨梅

四川大学锦城学院 计算机与软件学院 四川 成都 611731

**【摘要】**人机交互主要研究人和计算机以及二者之间关系的技术。通过对三维交互技术的深入研究,本文主要探究了视点漫游技术、三维几何变换、三维观察和坐标变换,采用 OSG (Open Scene Graph, 三维实时渲染引擎) 技术设计并实现了多相机多视图的三维场景漫游仿真,即操作多相机,用多个视图显示。最终实现多角度观察三维场景中的物体,给用户带来身临其境的体验。

**【关键词】**视点漫游; 三维交互; 多相机; 多视图; OSG

## 1 引言

人机交互一直是计算机领域重要的研究问题之一。三维人机交互技术不同于传统的二维窗口、图标、菜单等形式的图形交互技术,它主要研究人和计算机之间怎么构造出一种自然直观的三维交互环境<sup>[1]</sup>。三维交互技术须采用六自由度输入设备。对于六自由度,可以理解为:沿三维空间 X、Y、Z 轴做平移操作和旋转操作,而现在流行的用于桌面图形界面的交互设备,如鼠标、轨迹球、触摸屏等只有两个自由度(沿平面 X、Y 轴平移)。因此,新的交互方式的研究变得迫切。

## 2 关键技术

### 2.1 视点漫游技术

漫游(即变换视点)是三维世界中一种基础性的导航任务。视点漫游技术种类繁多,有驾驶技术、路径规划技术、集成在桌面环境下的相机控制<sup>[2,3]</sup>。结合这

些技术的优缺点,本文根据实际需求设计了由交互设备控制的自由式漫游,即用户通过外设(鼠标和键盘)指定漫游者的运动方向、速度、观察方向,从而对三维场景进行漫游观察。这种方式比较随意,用户有完全的自主性。

### 2.2 三维观察

对场景模型来说,三维观察就是世界坐标变到观察坐标。在计算机图形学中,我们可以在世界坐标系中通过建立观察坐标系来实现对三维物体的观察,建立观察坐标系。观察坐标系通过指定观察者的观察位置和投影平面的位置来设置参考系,在此基础上,将景物观察坐标变换到投影平面,得到相对于平面的新的坐标位置,再通过映射操作到输出设备。通过这个操作,使用可视面识别和表面绘制程序将观察范围内的物体正确的显示在设备视口上,而在观察范围外的物体将不被显示。

三维场景物体的显示流程如 1[4] 所示。

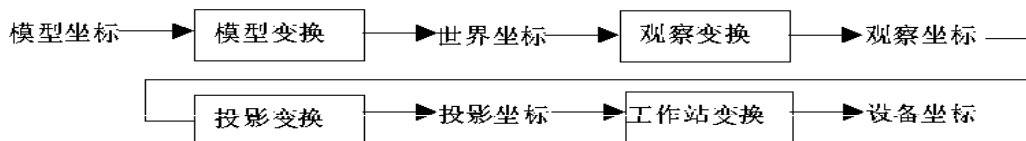


图 1-2 三维变换流水线

图 1 三维场景物体的显示流程

### 2.3 三维几何变换

三维几何变换分为旋转、缩放和平移。三维几何变换可以通过数学公式、三维齐次坐标及  $4 \times 4$  变换矩阵的乘积来表示。在二维基础上,增加 Z 轴参数,形成了三位坐标,从而实现三维的几何变换和建模。在三维坐标上,对移动距离设计三维变换向量对物体进行变换,或三个坐标系上的缩放因子对物体进行缩放。但是,对于三维旋转来说,三个坐标系上的扩展变得复杂,对于空间方向的任意性,可以通过给定轴的方向和旋转的角度一起构造一个旋转矩阵。

### 2.4 坐标变换

在三维空间创建和显示一个(多个)几何物体时,首先必须建立世界坐标系,然后,需要指定视点的方位、视线和成像面的方位,为了观察到物体的成像,还必须各坐标系之间视线变换之后,进行投影变换,才能得

到物体的成像。本文主要讨论了三维物体显示中的视点变换、模型变换、投影变换和视口变换。

#### 2.4.1 视点模型变换

合并视点变换和模型变换为视点模型变换,调整待拍摄物体对象的姿态、位置,以及相机对物体的校准。其中,视点变换主要针对相机的位置进行调整,即视点的方向和位置。在三维场景漫游中,改变相机的位置就改变了视点的观察方向和位置。经过一系列的转换,场景从世界坐标系(World Coordinate System, WCS)变换成相机坐标系(Viewing Coordinate System, VCS, 也叫视点坐标系)[5]。其中,任意行向量  $V_{wcs}$ (世界坐标系下)变换到相机坐标系的变换公式是:

$$V_{vcs} = V_{wcs} * M_{view}$$

### 2.4.2 投影变换

场景中的研究对象经过视点模型变换后, 将其显示在了预期位置上。三维物体在显示器上显示效果是二维的, 那么在原来三维物体上需要通过投影降低其维数。通过投影变换将视景内的物体以图像形式显示出来, 形成视景物, 多余部分将被裁减掉。投影包括透视投影和正射投影, 在三维场景中, 常使用透视投影, 即离视点越近, 观察到的物体越大, 反之物体越小。

透视投影变换过程中, 观察平面上的一点到观察体投影线上点的对应关系是比例变换关系, 则标准二维输入设备在显示器上点取得的一点  $S(x_s, y_s)$ 。经设备坐标正规化后, 可按比例关系在观察体底面上投影为点  $O(x_0, y_0, z_0)$ , 由计算机图形学相关理论可知, 透视投影的场景观察体由 7 视点到底面距离  $near$ 、到底面距离  $far$  及相机的 FOV (field-of-view) 三个参数唯一确定。因此观察体远裁剪面的四个顶点 ( $V_0, V_1, V_2, V_3$ ) 坐标可由当前视点的空间坐标求出。为了在观察坐标系下求得  $O$  点的坐标, 将观察平面上的设备坐标正规化。求出  $O$  点坐标后, 用视点坐标与  $O$  点坐标建立空间直线方程, 使用检测算法即可得到场景中距离视点最近的虚拟实体。视景物为观察平面和观察体底面之间的棱台体, 其内的对象就是“被看得见”的部分。

### 2.4.3 视口变换

视口是将视景物内投影的物体显示在二维的矩形视口平面上。这一变换意味着场景最终变换到了窗口坐标系, 该坐标系以屏幕窗口左下角为坐标原点,  $X$  和  $Y$  方向取值在  $(0, 0)$  到  $(width, height)$  范围内,  $Z$  方向取值范围为  $[0, 1]$ , 即 OpenGL 深度缓存 ( $Z$ -Buffer) 值的含义。

## 3 三维实时渲染引擎

OSG, 全称是 Open Scene Graph, 即三维实时渲染

引擎, 是开源的三位图形引擎, 具有高性能、易用等特点, 在虚拟仿真、虚拟现实、科学和工程可视化领域得到广泛使用。

### 3.1 相机类

对 OSG 描述的模型 - 视图矩阵设计了一个相机类 (`osg::Camera01`), 主要实现对相机的管理, 其方法包含各种变换操作, 包含的主要函数有: 视点变换、模型变换、投影变换和视口变换。

视点变换采用函数 `void setViewMatrixAsLookAt(const osg::Vec3d &eye, const osg::Vec3d &center, const osg::Vec3d &up)` 来设置视点的方向和位置, 即相机的观察位置, 这样使得使用者能根据需要调整视点, 对三维场景中的物体多方位观察。模型变换主要通过位置变换节点 (`PositionAttitudeTransform`) 和矩阵变换节点 (`MatrixTransform`) 类来实现模型变换。OSG 中使用函数 `void setProjectMatrixAsFrustum(double left, double right, double bottom, double top, double zNear, double zFar)` 来实现投影变换, 视口变换用函数 `void setViewport (int x, int y, int width, int height)` 来实现。

### 3.2 多视图类

多视图类 `osgViewer::CompositeViewer` 负责多个视图的管理及同步工作。该类包含了多个视图 (`osg::View`), 用来在场景中根据相机拍摄的内容设置多个视图来显示, 以便观察者从不同的角度对场景进行了解。

### 3.3 漫游器

在浏览整个三维场景时, 矩阵变换至关重要, 通过适当的矩阵变换可以获得各种移动或渲染效果。OSG 提供的三维场景漫游器采用实时修正场景相机 (`Camera` 类) 观察矩阵 (也就是观察者的观察位置和姿态) 的方式实现平滑的导航浏览。

假设相机的位置姿态矩阵为  $M_{world}$ , 观察矩阵为  $M_{view}$ , 则有:

$$M_{world} = M_{view}^{-1}$$

相机的位置姿态矩阵事实上可以理解为相机坐标系下的顶点向世界坐标系转换的变换矩阵, 而观察矩阵则为世界坐标系下的顶点转换到相机坐标系中。

## 4 多相机多视图漫游仿真

本文基于 OSG 技术, 设计并实现了多相机多视图的三维交互漫游, 即采用多相机从不同的角度观察场景, 将观察到的内容显示在不同视图中。根据相机原理, 设计了一系列变换矩阵, 利用 OSG 相关类和函数进行处理, 以及设计了三维交互漫游器, 即通过键盘按键或鼠标操作来控制相机的移动, 以改变观察点的位置和方向, 从而实现了多角度地观察三维场景, 便于使用者从三维场景中获取感兴趣的信息和全方位地观察场景。图 2 主要为类似于鹰眼全景视觉观察效果图和相机左右漫游观察场景的效果。



图 2 漫游观察效果图

### 结束语

三维场景的交互主要包括两种, 一种是通过交互实现场景的漫游, 即改变相机的观察方式; 另一种是通过交互改变场景中某个对象的位置姿态和形状。本文采用 OSG 技术, 设计了多相机多视图的三维交互漫游方式, 通过相机的控制改变观察方向, 并将不同位置观察到的场景内容显示在不同视图中。这种方式有利于用户通过

改变视点实现三维场景漫游,从不同角度观察三维场景,从而获得所感兴趣的信息。

**【参考文献】**

[1] 雷超,戴国忠. 三维交互体系结构的研究与实现[J]. 计算机研究与发展,2001.38(5):557~562.

[2] 永刚. 桌面环境下的三维用户界面和三维交互技术[D]. 研究中国科学院研究生院博士学位论文,2005.

[3] 程成. 虚拟环境人机交互技术研究[D]. 中国科学院研究生院. 博士学位论文,2002.

[4] 张全伙,张剑达. 计算机图形学[M]. 第一版. 北京:机械工业出版社,2004:20~22.

[5] 王锐,钱学雷. 三维渲染引擎设计与实践[M]. 北京:清华大学出版社. 2009.