

基于 Creator 和 OpenGVS 的三维仿真软件的开发

Development of 3D Simulation Software Based on Creator & OpenGVS

吴盘龙 (南京理工大学自动化学院 南京 210094)

李星秀 (南京理工大学理学院 南京 210094)

摘要: 采用虚拟现实建模软件 Multigen Creator 进行视景建模,视景管理软件 OpenGVS 来实现场景驱动,构建了一个比较完整和逼真的三维仿真实验系统,并介绍了用 OpenGVS 来作为实时场景三维动画的驱动来开发仿真软件中需要解决的几个关键问题,包括目标的建模、视点的选取,碰撞的检测,以及爆炸效果的产生。这些技术的运用使得软件仿真的效果更加逼真。

关键词: 虚拟现实 视景仿真 建模 碰撞检测

1 引言

一般的可视化仿真是基于 PC 机环境下的 3DMAX 和 OpenGL,前者用于实体的建模,后者实现对模型的驱动。随着计算机技术的飞速发展,人们对于可视化仿真的要求进一步提高,提出了虚拟现实(Virtual Reality, VR)的概念。虚拟现实是利用计算机生成一种模拟环境,通过多种传感设备使用户“沉浸”到该环境中,实现用户与该环境直接进行自然交互的技术。典型的 VR 系统的组成如图 1 所示。因此本文利用 Multigen Creator 建立仿真系统所必需的复杂三维模型,然后利用 OpenGVS 实现场景的三维动画驱动。

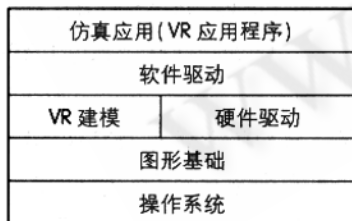


图 1 VR 系统结构图

2 系统建模

Multigen Creator 是一个功能强大的专业三维建模工具^{[1][3]},它为实时交互应用程序建立虚拟场景模

型如:飞机、导弹对象和地形,生成 OpenFlight 的图形数据格式文件(.flt)。OpenFlight 数据格式已成为虚拟现实业界的事实标准,它使用几何层次结构和属性来描述三维物体。

2.1 飞机、导弹三维模型的建立

(1) 飞机、导弹的基本结构。自然界的绝大多数景物为理想漫反射体,其模型称为 Lambert 漫反射模型,本文所组织的轰炸机、导弹三维模型即为此类模型,且轰炸机、导弹的有些部件是对称的。

(2) 细节度(LOD)的运用。LOD 可使模型在较远的地方比较粗糙,数据量较小,但又不影响真实感;在较近的地方,真实感则会大大加强。在设置的变化距离时,可根据实际而定。LOD 的选项用到 Morph 功能,以使得在远近之间的切换时,变化比较平缓,而不会出现突变。

(3) 自由度(DOF)的运用。DOF 用来在数据库中定义模型某一部分的移动和旋转。在进行状态转变时,可用 Switch 来控制,Switch 本身可以按照顺序控制模型,在 OpenGVS 程序设计中方便对其控制,便于在交互式虚拟现实中的应用。

(4) 纹理贴图。纹理贴图是实现场景真实感的一个重要途径,这些纹理来自网络上下下载的图片,经过 Windows 的画图或 PhotoShop 等工具编辑处理后,即可作为飞机的纹理。但每张纹理图片的大小都是 2 的

n 次方。

2.2 三维地形模型的生成

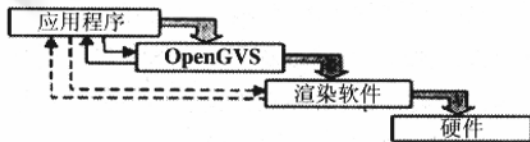
系统所采用的地形数据来源于美国国防部国家测绘局 NIMA, 原国防部制图局 DMA 的数字地形高程数据 (Digital terrain elevation data, DTED)。DTED 由地形高度矩阵组成, 提供地球表面的起伏信息。

当模型应用于交互式虚拟场景中, 在确保真实感的前提下, 应尽量减小数据量, 在 MultiGen Creator 中提供了 Vsimplify 功能, 它可在保证真实性的前提下对模型进行简化。

3 用 OpenGVs 实现三维动画系统设计

3.1 OpenGVs 简介

OpenGVs 是 Quantum3D 公司的产品, 用于视景仿真软件的实时开发, 易用性和重用性好, 有良好的模块化、编程灵活性和可移植特性^[2]。OpenGVs 的体系结构如图 2 所示。OpenGVs 的 API 分为视点、通道、烟雾、帧缓存、几何体、光源、对象、视景、工具、特效等各种资源, 开发者可以按照程序的需要调用这些资源来驱动硬件实时产生所需的图形和效果。



- 应用程序对 OpenGVs 函数的调用与返回
- 对底层渲染程序调用与返回 (例如调用 OpenGL、3Dfx、Glide、Direct 3D 等)

图 2 OpenGVs 的体系结构

3.2 OpenGVs 接口及视点的设置

OpenGVs 是建筑于多种 3D API 的渲染机制基础之上的, 采用 OpenGVs 开发动画软件系统时首先必须设置应用程序与 OpenGVs 软件之间的接口, 主要有帧缓存 (framebuffer)、视景 (scene)、视点 (camera)、通道 (channel) 以及光源 (lightsource) 等, 完成资源的连接后就可以向视景加入用户定义的对象。用 OpenGVs 实现为:

```
GV_scn_add_light( scene, light );
GV_chn_set_camera( channel, camera );
GV_chn_set_scene( channel, scene );
```

```
GV_fbf_add_channel( fbf, channel );
```

在光电干扰的仿真中, 选择合适的视点显得尤为重要。因为, 如果选择了一个好的视点, 那么就能使人们能够实时地看到导弹和飞机所作的各种动作, 以及定性的看到干扰效果, 使仿真显得更加逼真。OpenGVs 提供了能够控制视点的一个非常好的资源——相机 (Camera)。

每一个相机都有一个或多个平台, 创建相机之后, 这个相机就自动包含一个平台, 编号记做 0, 对于简单的应用, 这就已经足够了。但是对于更复杂的视景需要, 就要创建更多的平台, 编号依次为 1、2、3……。每一个平台都有自己的位置和转角, 但是每一个平台都是相对于编号比自己小一的那个平台定义的。而编号为零的那个平台的缺省位置就是世界坐标系的 {0, 0, 0}, 也可以由人们自己设置。本文的相机创建两个平台, 编号为 0 的基平台绑定在飞机上, 编号为 1 的平台, 相对于上一个平台定义位置, 使观察者能够较好的看到逼真的场景^[4]。用 OpenGVs 实现为:

```
#define BASE_PLATFORM0
#define EYEPOINT_PLATFORM 1
GV_Camera camera;
GV_Obi obi;
static G_Position eye_offset = { 0.0, 20.0, 100.0 };
};
GV_cam_create( &camera );
GV_cam_set_name( camera, "VIEWER" );
GV_obi_inq_by_name( "AIRCRAFT", &obi );
GV_cam_set_mount_obi( camera, obi );
GV_cam_add_platform( camera );
GV_cam_set_position( camera, EYEPOINT_PLATFORM, &eye_offset );
```

其中 "BASE_PLATFORM" 和 "EYEPOINT_PLATFORM" 是定义的相机的两个平台。"eye_offset" 的位置是根据不同的观察者来设定的, 力求达到最好的观察效果。导弹追踪飞机的场景如图 3 至图 5 所示。

3.3 模型的调用及其控制

OpenGVs 支持 Terrex TerraPage、Multigen OpenFlight、Wavefront obj 和 3D Studio (3DS) 等模型数据格式。以轰炸机的调用为例, 首先使用写字板编辑一个名为 gvsinit. gvc 的文本文件, 在 gvsinit. gvc 中加入语

句: `Import file = bomber. flt name = BOMBER`。在 OpenGVS 显示的语句为:

```
Static GV_Obi bomber;
GV_obi_instance_by_name("BOMBER",
&bomber);
GV_scn_add_object(scene, bomber);
```

导弹和地形在视景中的显示方法同上,不同的是要设置各自的属性,包括位置和姿态等。

在 OpenGVS 中是运用仿真回调来实现控制对象运动的。对于这种类型的仿真需求一般的用户须定义对象(实例)以包含用户仿真回调函数。随后运行期间,OpenGVS 在合适时间和位置调用用户函数。完成调用后函数返回给 OpenGVS, OpenGVS 继续其他处理任务。要建立仿真回调函数,就必须调用“GV_obd_set_sim_callback”或“GV_obi_set_sim_callback”。通常,需要为对象定义建立回调函数,这样所有此定义的实例将包含此回调引用。此后,当对象实例打开时(GV_obi_set_state 为 G_ON), OpenGVS 在运行期间的每一帧调用指定的用户函数。用户可以自己设置导弹的运动形式(本文将导弹设置成了运用比例导引法追踪飞机),将它编为程序添加到上述回调函数中。同理也可以为飞机加入回调函数,控制飞机的运动。

3.4 碰撞检测

碰撞检测是计算机仿真中的一个重要的问题。导弹在飞行过程中,对每一帧动画都要进行和视景对象的碰撞检测,以判断导弹是否击中目标,从而看出飞机对导弹干扰的效果^[5]。

碰撞检测时获取导弹前一帧和当前帧的位置,调用函数 `GV_geo_inq_intersection()`。它用来检测视场中的对象实例和用户指定的线段是否发生相交。检测导弹是否打中飞机,即与飞机发生碰撞;以及防止飞机飞得过低而钻入地下等。如果发生碰撞,碰撞点在大地中的位置。用 OpenGVS 实现为:

```
extern G_Posotion p0, p1;
static G_Mask exclude = 0;
static GV_Geo_isc_control_mask;
type_mask = GV_GEO_ISC_MASK_SERCH_FACE;
static GV_Scene scene = NULL;
GV_Geo_isc_data data;
if(! scene)
```

```
GV_scn_inq_by_name("OTW", &scene); GV_geo_
_inq_intersection(scene, &p0, &p1, type_mask, exclude,
&data);
```

通常我们不用函数 `GV_geo_inq_intersection()` 的返回值来确定碰撞是否发生,而是通常用结构 `GV_Geo_isc_data` 中的成员 `isc_result` 来确定碰撞的发生。

3.5 爆炸效果的生成

在仿真软件的演示过程中,为了更加真实地反映光电干扰的效果,并在视觉上达到更强的真实感,我们加入了飞机在光电干扰过程中投出的干扰弹的爆炸效果。在 OpenGVS 中,这些特殊效果可由纹理映射、纹理序列动画等方法生成。

OpenGVS 纹理映射的方法与在 OpenGL 中采用的方法基本相同。导弹和干扰弹爆炸过程中产生的火焰和烟由浓到淡直至消失的过程可以通过改变纹理的透明度来实现的。调入一张纹理图片具体的 OpenGVS 实现为:

```
GV_txr_create(&texture);
GV_txr_set_filename(texture, "texture/explosion_
cartoon. rgba");
GV_txr_set_name(texture, "CARTOON_EXPLO-
SION");
GV_txr_define(texture);
```

动画序列创建完之后就可以在需要时加入到视景中。并设置纹理的透明度,使透明度由浓到淡,用来显示爆炸时火光和烟的慢慢消失的效果,还可以设置显示的爆炸范围。爆炸效果如图 6 示。



图 3 OpenGVS 中的飞机飞行场景图

4 结论

本文提供一种基于 PC 机平台,使用 Multigen Cre-

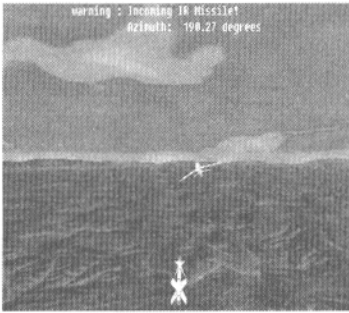


图 4 导弹追踪飞机的场景 1

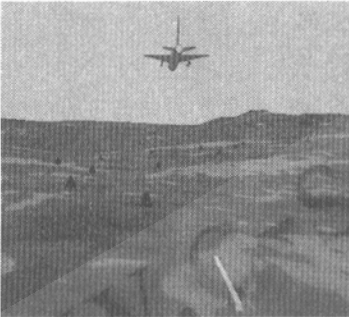


图 5 导弹追踪飞机的场景 2

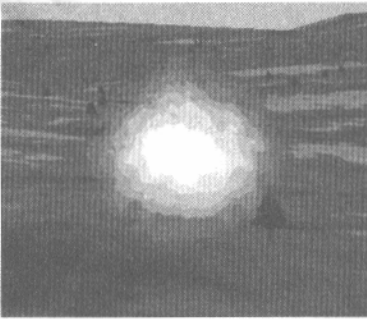


图 6 爆炸的效果图

- 3 吴家铸等, 视景仿真技术及应用[M], 西安: 西安电子科技大学出版社, 2001.
- 4 谢薇、郭齐胜、郑长伟等, 基于 OpenGVS 的视景仿真的关键技术研究[J], 计算机仿真, 2001, 18(6): 26-28.
- 5 吴春波、张靖, 导弹模拟训练器动画的 OpenGVS 实现[J], 弹箭与制导学报, 2002, 23(1): 13-16.

ator 和 OpenGVS 进行三维仿真系统开发的方案, 并对系统开发中所用到的关键技术进行了分析。解决了上述的关键问题之后, 动画效果更加逼真, 使人们的视觉感受更接近于战场上的实际感受。系统结构具有较好的通用性, 对开发类似系统有一定的借鉴作用。

参考文献

- 1 Multigen - Paradigm Inc. Multigen Creator User's Guide version 2.6[Z]. USA, 2003.
- 2 Quantum 3D Inc. OpenGVS Programming Guide Version 4.4[Z]. USA, 2001.