1　绪　论

1.1 问题的提出及研究意义

房地产业的竞争越来越激烈，地理位置、社区规模、环境营造以及户型设计的比拼也异常火爆。楼盘在 规划设计 和空间创意方面，已经难有革命性的突破。很多开发商还局限在楼书、条幅等相对传统的表现手法上。购房者的日益成熟，他们需要更直观、更快捷、更全面的了解楼盘本身的信息，传统的 楼书、条幅、等无法完全满足他们的需求。

基于 VEGA 软件的楼盘演示技术一种全新的房地产宣传模式。它可以将动画、视频、三维立体、文字等数字资源整合在一个交互式的整体中，以图文并茂、生动活泼的动态形式表现出来。客户可以全面立体，逼真的效果展示出楼盘外貌、小区风景、可以设定楼成朝向面积首期金额等条件，迅速找到适合自己的最佳居室，并能为房地产商有效提高工作效率，降低建筑成本，与传统的平面楼书相比，电子楼书具有不受篇幅的限制，制作成本低廉。不受场地限制，便于携带、方便保存。

1.2 本课题研究领域国内外的研究动态及发展趋势

1.2.1 虚拟实现技术在国内外研究现状与发展

（1） 国外的研究状况

虚拟现实技术在国外相对发展的比较早，基本上分为三个阶段。

第一阶段，从 20 世纪 50 年代到 70 年代。这一阶段，虚拟现实技术并没有形成完整的概念。

第二阶段，20 世纪 80 年代初到 80 年代中期。该阶段开始形成虚拟现实技术的基本概念。

第三阶段为虚拟现实技术全面发展阶段。进入 90 年代，迅速发展的计算机硬件技术与不断改进的计算机软件系统相匹配，使得基于大型数据集合的声音和图像的实时动画制作成为可能；人机交互系统的设计不断创新，新颖、实用的输入输出设备不断地进入市场。而这些都为虚拟现实系统的发展打下了良好的基础。例如 1993 年的 11 月，宇航员利用虚拟现实系统成功地完成了从航天飞机的运输舱内取出新的望远镜面板的工作，而用虚拟现实技术设计的波音 777 获得成功，是近年来引起科技界瞩目的又一件工作。可以看出，正是因为虚拟现实系统极其广泛的应用领域，如娱乐、军事、航天、设计、生产制造、信息管理、商贸、建筑、医疗保险、危险及恶劣环境下的遥操作、教育与培训、信息可视化以及远程通讯等，人们对迅速发展中的虚拟现实系统的广阔应用前景充满了憧憬与兴趣 。

（2）国内的研究状况

和一些发达国家相比，我国 VR 技术还有一定的差距，但已引起政府有关部门和科学家们的高度重视。根据我国的国情，制定了开展 VR 技术的研究，例如：

北京航空航天大学计算机系是国内最早进行 VR 研究、最有权威的单位之一，他们首先进行了一些基础知识方面的研究，并着重研究了虚拟环境中物体物理特性的表示与处理；在虚拟现实中的视觉接口方面开发出了部分硬件，并提出了有关算法及实现方法；实现了分布式虚拟环境网络设计，建立了网上虚拟现实研究论坛，可以提供实时三维动态数据库，提供虚拟现实演示环境，提供用于飞行员训练的虚拟现实系统，提供开发虚拟现实应用系统的开发平台，并将要实现与有关单位的远程连接。

浙江大学 CAD&CG 国家重点实验室开发出了一套桌面型虚拟建筑环境实时漫游系统，该系统采用了层面迭加的绘制技术和预消隐技术，实现了立体视觉，同时还提供了方便

的交互工具，使整个系统的实时性和画面的真实感都达到了较高的水平。另外，他们还研制出了在虚拟环境中一种新的快速漫游算法和一种递进网格的快速生成算法。

清华大学计算机科学和技术系对虚拟现实和临场感的方面进行了研究,例如球面屏幕显示和图像随动、克服立体图闪烁的措施和深度感实验等方面都具有不少独特的方法。他们还针对室内环境水平特征丰富的特点，提出借助图像变换，使立体视觉图像中对应水平特征呈现形状一致性，以利于实现特征匹配，并获取物体三堆结构的新颖算法。

中科院威海分院主要研究虚拟现实中视觉接口技术，完成了虚拟现实中的体视图像对算法回显及软件接口。他们在硬件开发上已经完成了(Liquid Crystal Display ,LCD）红外立体眼镜，并且已经实现了商品化。

北方工业大学 CAD 研究中心是我国最早开展计算机动画研究的单位之一，中国第一部完全用计算机动画技术制作的科教片《相似》就出自该中心。

另外，还有一些社会团体公司也在进行虚拟现实技术应用的研究。

1.2.2 虚拟现实技术在国内外房地产中的研究现状与发展

虚拟现实 技术的研究和 发展，广泛地应用于 房产销售，目前广州、上海、北京的房地产发展商都采用了该系统， 国外的加拿大、美国等经济和科技发达的国家都非常热门，是当今房地产行业一个楼盘档次、规模和实力的象征和标志，其最主要的核心是房地产销售!

1.3 本文研究的目的和内容

1.3.1 研究的目的

房地产虚拟演示系统是最佳的楼盘销售演示方式。可以取代样板房，而价格远远低于制作样板房。在楼盘建设完工之前，虚拟演示系统就能把楼盘及其周围的环境和道路

都虚拟出来，能全面、详尽的体现楼盘的优势。基于虚拟现实技术的楼盘演示系统的研究与开发有非常重要的意义。其一：购楼者可方便获取楼盘、小区、甚至每个房间的信息，找到适合自己的单元。其二：为房地产商有效提高工作效率，降低建筑成本。

1.3.2 研究的内容

由于要完善房地产楼盘虚拟现实系统的工作量及技术要求是非常高的，所以本课题主要应用现已成熟的虚拟现实技术理论，针对房地产的特点，以西安科技大学校园环境为例，研究实现房地产虚拟现实系统计算机立体模型可控可视化的技术方法、实现途径、开发思路和技术路线，包括系统平台构建、技术处理、软件平台、模块构建、界面开发及实时交互等，具体章节安排如下：

（1）绪论

这一部分主要说明这几个问题：问题的提出、课题研究的意义、国内外研究现状、本文研究的目的和内容等。

（2）虚拟现实理论与技术

主要介绍虚拟现实的概念、特征，立体显示的原理、三维图形的变换，虚拟现实系统的构成、分类及虚拟现实技术的应用。

（3）系统的建模与驱动

介绍房地产 VR 系统的开发工具，包括建模软 Multigen Creator、实时视景模拟驱动软件 Vega 及图像预处理软件 Photoshop 等。

（4）开发房地产 VR 系统的关键技术

在开发房地产 VR 系统的过程中，运用纹理映射、模型优化及场景优化技术，包括面合并、删除冗余面、LOD 技术、Instance 技术、环境效果等。

（5）房地产 VR 系统的实现

对系统进行了总体的分析和设计，介绍了房地产 VR 系统的实现技术路线和方法，并介绍了 Vega 应用程序建立的实现方式。

（6）结论与展望

1.4 系统的研究技术路线

虚拟现实技术的核心是建模与仿真，就建模和仿真的本质而言，它是对真实物理系统在某一层次上的抽象，与实际的物理系统相比，用户在这个抽象模型上可以更高效、更节省、更灵活、更安全地对物理系统进行了解和设计。为此，本论文就建模和驱动两个方面来讨论，实现工具选择视景建模与仿真工具 MG1tiGenCreator/Vega。该工具是由处于世界领先地位的视景仿真技术公司 Multigen-Paradigm 公司提供的。具体的实现方案如图 1.1 所示。 具体实现步骤与内容如下：

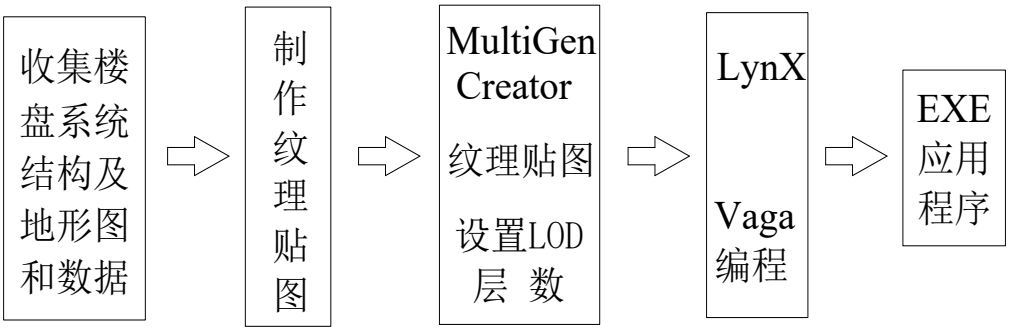


图 1.1 利用 MultiGen Creator/Vega 实现系统

1.4.1 收集资料数据

建造模型的首要工作是收集模型的数据资料，考虑到井工开采虚拟现实系统的实用性和复杂性特点，在进行建模前，必须要确定系统场景的实体几何尺寸，相互位置坐标，模型相似精度。这就需要进行详细的资料数据收集，如果收集的资料不全或者数据偏差较大，必然影响到后期模型的建立，进而影响到该仿真场景与实际场景的可靠性和准确度，这就要求不但要收集到最新的、最准确的资料信息，而且要到实地进行考察。

1.4.2 制作纹理贴图

在房地产虚拟现实系统中，由于模型精细程度的限制，场景的真实感很大程度上要靠纹理来体现。由于直接在 Creator 中不能对纹理图片进行全面细致的处理，要借助其他系列图像处理软件来完成。素材收集时用数码相机拍的实景照片，借助于 Photoshop 7.0 图像处理软件进行纠正处理，纹理图片的长度和宽度全部限制为 2 的幂次单位像素大小，否则进行纹理贴图时就会丢失或倾斜变形，然后利用插件存储为 RGB 或 RGBA 格

式，作为模型纹理库，供 Creator 调用。

1.4.3 建立实体模型

房地产虚拟现实系统场景模型是整个实时视景仿真的核心。模型的好坏，直接影响到运行的效果和场景的逼真度。根据房地产系统建模的需求分析，不同模型有着不同的重要性，同时当前的模型也有着不同的建模要求，因此在建模的过程中，对不同的模型给予区别对待。几何模型及场景建完后，给模型的表面加上各种材质，设定各种光照条件，进行几何表面纹理贴图，接着要把初步模型加入虚拟现实系统进行验证，不断修改使得模型看上去更加光滑、真实、有质感，使得模型更加符合实际要求，成为最终模型，生成(*.flt)格式模型文件。

对于一些复杂的模型，Creator软件的建模功能没有3Dmax或者Maya功能强大，所以，虽然Creator的数据格式与Lynx兼容，但在必要的时候也需要借助其它软件来辅助建模，然后将其模型格式转化为Creator所识别的格式，再进行修改，具体格式转化方法将在后面章节介绍。

1.4.4 系统场景合成

运用 Vega 的 Lynx 功能提供图形用户界面，创建用于实时应用的 ADF(Application Definition Files)文件，在图形用户界面下，修改参数，最终生成（*.ADF）格式文件，为开发出强大的实时应用程序作准备。

1.4.5 Vega 应用程序的实现

在 ADF 文件之外，通过编写程序来实现房地产楼盘虚拟现实系统的实时性和交互性需求。在 NT 环境下，选择 Vega 的基本开发环境为 VC＋＋，可以很方便地和 C/OpenGL 相结合，从而方便、快捷地实现系统的友好界面和功能扩展编程，将 ADF 文件和扩展用户程序载入到 Vega 程序中，进行实时应用编译，最终形成房地产楼盘虚拟现实系统。

如图 1.2 所示：

2　虚拟现实的理论与技术

2.1 虚拟现实的理论基础

2.1.1 虚拟现实的概述

虚拟现实技术被认为是九十年代高新技术的尖兵。有关人士认为，80　年代是个人计算机的年代，90 年代是多媒体计算机的年代，21 世纪初将是虚拟现实技术的时代。虚拟现实技术刚一出现，就立即受到了高度重视，从来没有一项新技术这样引人注目。虚拟现实（Virtual Reality，简称 VR），源于美国。这一名词是由美国 VPL Research 公司创始人拉尼尔（Jaron　Lanier）在 1989 年提出的，我国著名科学家钱学森将它翻译为灵境技术。它与传统的模拟技术完全不同，是将模拟环境、视景系统和仿真系统合三为一，并利用头盔显示器、图形眼镜、数据服、立体声耳机、数据手套及脚踏板等传感装置,把操作者与计算机生成的三维虚拟环境连结在一起。操作者通过传感器装置与虚拟环境交互作用，可获得视觉、听觉、触觉等多种感知，并按照自己的意愿去改变“不随心”的虚拟环境。从而使得在视觉上产生一种沉浸于这个环境的感觉，可以直接观察、操作、触摸、检测周围环境及事物的内在变化，并能与之发生　“交互”作用，使人和计算机很好地“融为一体”，给人一种“身临其境”的感觉。虚拟现实技术应用范围非常广泛，与 VR 有关的领域是各种各样的，其中包括航空、航天、船舶、铁道、建筑、土木、科学可视化、医疗、军事、教育、娱乐、通信、艺术、体育等。

2.1.2 虚拟现实技术的特征

虚拟现实技术是一种高度逼真的模拟人在自然环境中视、听、动等行为的人机交互技术。它提供了一种先进的人机界面，最大程度地方便了用户的操作，其效率主要由系

统的沉浸程度与交互程度来决定。虚拟现实的特征，可以形象地用一个“3I”图加以描述。它们是　Immersion（沉浸性）、Interaction（交互性）、Imagination（想象性），如图 2.1 所示。

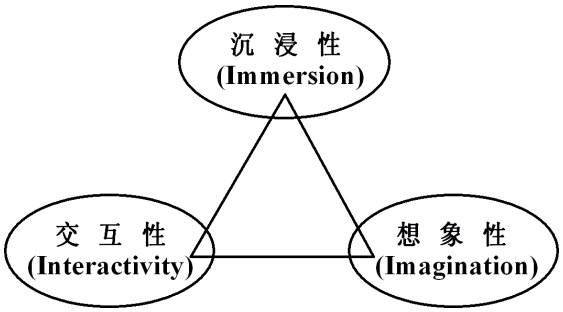


图 2.1 虚拟现实技术的基本特征

（1）沉浸性：沉浸性又称浸入性，是指用户感觉到好像完全置身于虚拟世界之中一样，被虚拟世界所包围。

（2）交互性：在虚拟现实系统中，交互性的实现与传统的多媒体技术有所不同，在传统的多媒体技术中，人机之间的交互工具从计算机发明到现在，主要通过键盘和鼠标进行一维、二维的交互，而虚拟现实系统强调人与虚拟世界之间要以自然的方式进行交互，如人的走动、头的转动、手的移动等，通过这些，用户与虚拟世界进行交互。虚拟现实技术的交互性具备三个特点：a.虚拟环境中人的参与与反馈；b.人机交互的有效性；c.人机交互的实时性。

（3）想象性：想象性指虚拟的环境是人想象出来的，同时这种想象体现出设计者相应的思想，因而可以用来实现一定的目标。所以说虚拟现实技术不仅仅是一个媒体或一个高级用户界面，它同时还可以是为解决工程、医学、军事等方面的问题而由开发者设计出来的应用软件，通常它以夸大的形式反映设计者的思想，虚拟现实系统的开发是虚拟现实技术与设计者并行操作，为发挥它们的创造性而设计的。

2.1.3 虚拟现实系统的构成

一般的虚拟现实系统都有 5 个关键部分组成：虚拟环境、VR 软件、计算机、I/O 设备、用户。用户通过输入设备直接对虚拟环境操作，并得到实时三维显示和其它的反馈信息，系统构成如同 2.2 所示。

（1）虚拟环境：提供一个交互的场景，可以从任意角度连续地观看和考察视景的变化，它一般是一个包含三维模型或环境定义的数据库。

（2）VR 系统：提供实时观察和参与虚拟世界的能力。

（3）计算机：VR 现实技术的基础设备，用来生成各种虚拟现实的场景，进行各种实时计算和处理。

（4）I/O 设备：用于观察和现实虚拟的场景，包括头盔显示器、立体眼镜、 鼠标、定位跟踪器等。

（5）用户：用户指虚拟环境的感受者，一般指人，整个虚拟场景的实时交互都是用户通过 I/O 设备来完成的。

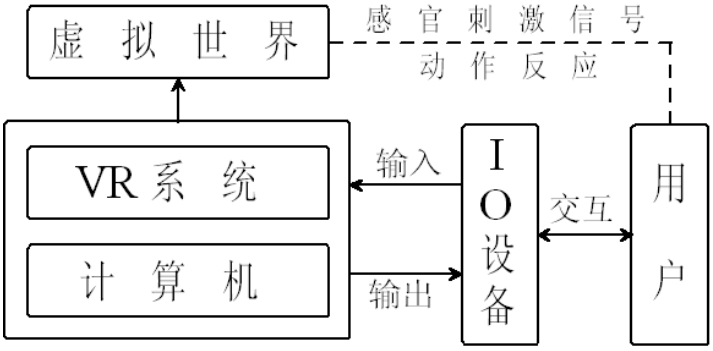


图 2.2　虚拟现实系统的构成

2.2 虚拟现实的三维图形变换技术与立体显示技术

在三维场景中经常出现数量大的、重复的几何体，这些几何体有时在观察者看来，只是在场景中的位置、方向或者大小、比例不同而已，在这种情况下，为了优化模型数

据库，就可以实现共享一个模型数据，通过图形的几何变换得到不同的几何体。

在计算机图形学中，设三维空间的任意一点 P(x,y,z)用 1×4 的齐次坐标矩阵表示为(x y z 1)，于是通过几何变换后的点 $P_1(x_1,y_1,z_1)$用 1×4 齐次坐标矩阵可表示为

$$(x_1,y_1,z_1)=(x\ y\ z\ 1)\cdot T \tag{2-1}$$

其中 T 是一个 4×4 矩阵，形如

$$T=\left[\begin{array}{ccc:c} a & b & c & p \\ d & e & f & q \\ h & i & j & r \\ \hdashline l & m & n & s \end{array}\right] \tag{2-2}$$

令

$$T_1=\begin{bmatrix} a & b & c \\ d & e & f \\ h & i & j \end{bmatrix} \tag{2-3}$$

$$T_2=\begin{bmatrix} l & m & n \end{bmatrix} \tag{2-4}$$

那么矩阵 T_1 将产生比例、对称、错切、旋转变换，T_2 产生沿三个轴向的平移变换。

（1）比例变换

$$T=\begin{bmatrix} a & 0 & 0 & 0 \\ 0 & e & 0 & 0 \\ 0 & 0 & j & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2-5}$$

其中 a、e、j 分别是 x、y、z 的比例因子,对于不同的比例因子将产生不同的比例效果，若 a、e、j 相同，则三个方向扩大或缩小的比例相同，如图 2.3 的（a）所示；若 a、e、j 不相同，则三个方向扩大或缩小的比例随相应比例因子而变，如图 2.3 的（b）所示。

(a) (b)

图 2.3　三维几何体的比例变换

（2）对称变换

对称变换包括对坐标原点、坐标轴及坐标平面的对称变换，三种变换原理相同，下面介绍关于 xoy，yoz，xoz 坐标平面的对称变换。

① 关于 xoy 平面对称的变换矩阵

$$T=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & -1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \tag{2-6}$$

② 关于 yoz 平面对称的变换矩阵

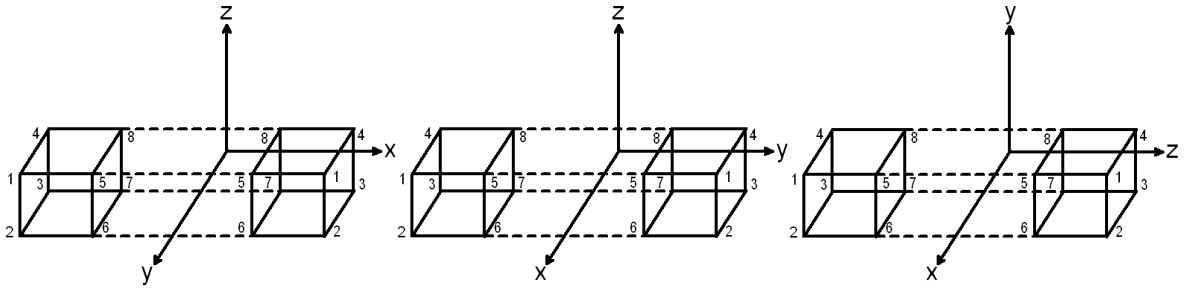
$$T=\begin{bmatrix}-1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \tag{2-7}$$

③ 关于 xoz 平面对称的变换矩阵

$$T=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & -1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \tag{2-8}$$

从三种变换矩阵可以看出，几何体针对不同平面对称变换时，只有一个坐标改变，其它两个坐标均不变，如图 2.4 的（a）、（b）、（c）分别所示为关于 xoy、yoz、xoz 平

面对称变换的立体图形。



(a)　　(b)　　(c)

图 2.4　三维几何体的对称变换

（3）错切变换

$$T=\begin{bmatrix} 1 & b & c & 0 \\ d & 1 & f & 0 \\ h & i & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2-9}$$

式（2-9）为沿 x，y，z 三个方向产生错切位移，其中

① 当 b=c=f=i=0 时，几何体沿 x 方向产生错切；

② 当 c=d=f=h=0 时，几何体沿 y 方向产生错切；

③ 当 b=d=h=i=0 时，几何体沿 z 方向产生错切。

（4）旋转变换

旋转一几何体有几种方法，这里只考虑绕某一坐标轴以及绕空间任一条直线的旋转。一个几何体的刚体旋转 T 必须满足下列条件：

a.物体上点与点、斜面与斜面之间的相对距离保持不变；

b.建立的坐标系是右手坐标系；

c.右手规则的约定用于确定旋转角的正负号，如图 2.5 所示，

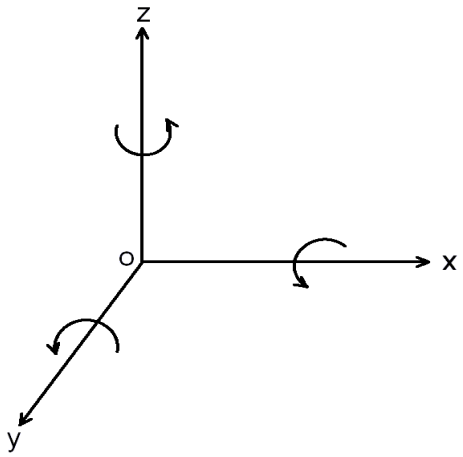


图 2.5 旋转角度正负号的确定

5　绕 x 轴旋转θ角

几何体绕 x 轴旋转时，x 坐标不变，y、z 坐标变换，变换矩阵为：

$$T_x=\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & \cos\theta & \sin\theta & 0\\ 0 & -\sin\theta & \cos\theta & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \tag{2-10}$$

5　绕 y 轴旋转θ角

几何体绕 y 轴旋转时，y 坐标不变，x、z 坐标变换，变换矩阵为：

$$T_y=\begin{bmatrix}\cos\theta & 0 & -\sin\theta & 0\\ 0 & 1 & 0 & 0\\ \sin\theta & 0 & \cos\theta & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \tag{2-11}$$

5　绕 z 轴旋转θ角

几何体绕 z 轴旋转时，z 坐标不变，y、x 坐标变换，变换矩阵为：

$$T_z=\begin{bmatrix}\cos\theta & \sin\theta & 0 & 0\\ -\sin\theta & \cos\theta & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \tag{2-12}$$

（5）平移变换

当几何体在空间进行平移变换的时候，几何体本身的形状河大小保持不变，只改变

空间位移。

$$T=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ l & m & n & 1\end{bmatrix} \tag{2-13}$$

其中，l、m、n 分别代表沿 x、y、z 方向的平移量。

2.3 虚拟现实的应用

VR 技术的应用极为广泛，Helsel 与 Doherty 在 1993 年对全世界范围内已经进行的 805 项 VR 研究项目作了统计，结果表明：目前在娱乐、教育及艺术方面的应用占据主流，达 21.4%；其次是军事与航空，达 12.7%；医学方面达 6.13%，机器人方面占 6.21%；商业方面占 4.96%；另外在可视化计算、制造业等方面也有相当的比重。

由于虚拟现实在技术上的进步与逐步成熟，其应用在近几年发展迅速，应用领域已由过去的娱乐与模拟训练发展到包括航空、航天、铁道、建筑、土木、科学计算可视化、医疗、军事、通讯等广泛领域。 随着 VR 技术的进一步成熟，其应用领域和应用深度将不断发展，其部分应用成果如组图 2.7～2.11 所示。

图 2.7 军事航空航天与 VR 项目开发（来自 http://www.paradigmsim.com）

图 2.8　建筑设计与 VR 项目开发（来自 http://www.pcvr.com.cn）

图 2.9　三维地理信息系统与 VR 项目开发（来自 http://www.paradigmsim.com）

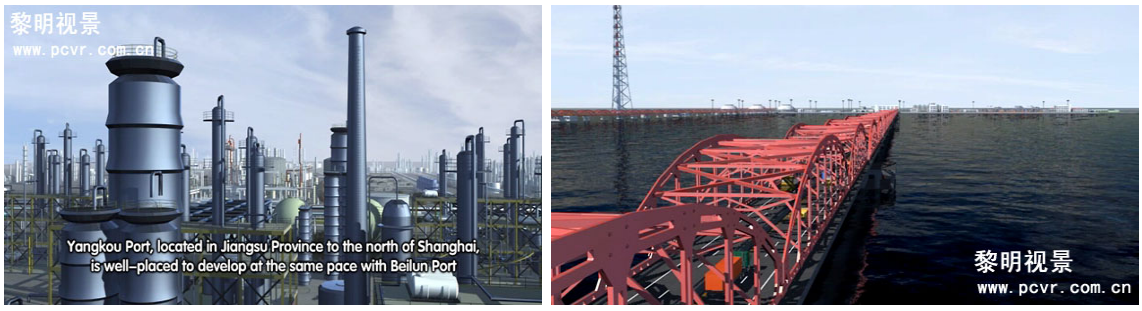


图 2.10　大型工程与 VR 项目开发（来自 http://www.pcvr.com.cn）

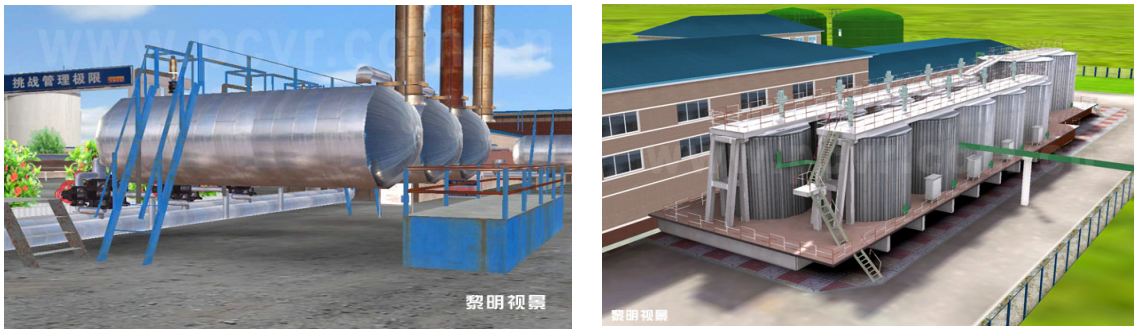


图 2.11　生产加工过程与 VR 项目开发（来自 http://www.pcvr.com.cn）

2.4 本章小节

本章介绍了虚拟现实的相关概念、虚拟现实的三维图形变换技术与立体显示技术，最后阐述了虚拟现实技术在各个行业领域的应用及成就。这些技术的研究和分析为本文的研究提供了理论保证。

3 系统的建模与驱动

3.1 实时仿真建模软件

目前，虚拟现实开发软件发展迅速，种类较多，使得很多现实场景可以通过诸如：Director、3Dmax、等程序软件来做模拟的演示和操作。但是，由于房地产楼盘个别系统构成和场景环境比较复杂，上述软件的自身局限性导致了模拟实验的仿真程度和实时交互性还不是很高，还不能达到一些较高标准虚拟场景的要求，下面就作者在虚拟现实系统所使用的工具软件做一简要介绍。

3.1.1 Multigen Creator 简介

MultiGen Creator 是 MultiGen-Paradigm 公司专门针对可视化仿真行业应用特点推出的实时可视化三维建模软件，它主要的特点在于其独创的用于描述三维虚拟场景的层次化数据结构—OpenFlight 数据结构。基于对实时应用优化的 OpenFlight 数据格式，MultiGen 提供了强大的多边形建模、矢量建模以及大面积地形精确生成等功能，配合多种专业可选模块及插件，用户能够高效地生成实时三维模型数据库，并与后续的实时仿真软件紧密结合。这些特点使得 MultiGen 系列软件在视景仿真、模拟训练、交互式游戏及工程应用、科学可视化等实时可视化仿真领域都有着广泛的应用[24]，MultiGen Creator 与 CAD 等其他建模软件不同，它主要考虑如何生成逼真的大面积地形、地貌等地理环境模型，以及如何提高模型的实时性。MultiGen Creator 还提供了其他的数据格式转换工具，如 Alisa/Wavefront、AutoCAD、3D Studio、Photoshop 等。它还具有动态重组数据库、生成动态仪表、生成实时地形等的功能，其软件初始运行界面如图 3.1

所示。

3.1.2 OpenFlight 数据格式

三维建模是虚拟现实系统设计的核心问题之一，而三维数据库又是整个建模的基础，三维数据库的组织逻辑（格式）对虚拟现实系统的运行质量有着极大的影响，常用的 CAD 格式是不适合实时系统的。因为对于针对虚拟现实应用而创建的模型，不仅仅要像普通的三维模型那样具有完整的几何外观，更为重要的是它必须具备一些能够满足实时应用需要的特质。比如模型在空间上各个独立模型元素之间的相对位置、层次关系、模型单元本身的一些属性和性质，以及组成模型的部分元素之间的相互关系和层次结构等重要信息。

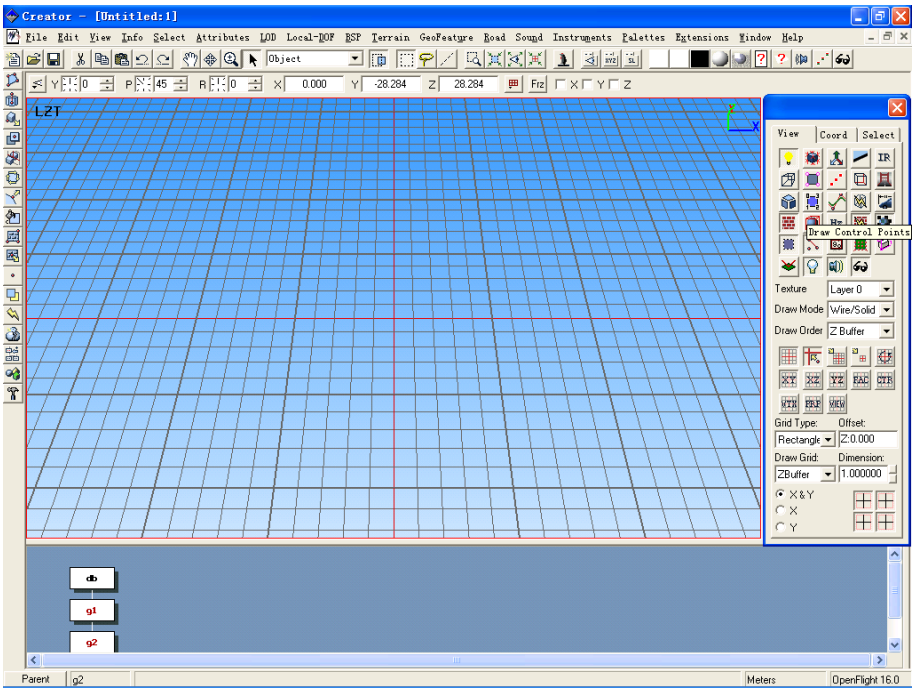


图 3.1 Creator 建模环境初始界面

OpenFlight 格式（*.flt）的模型数据库正是为了完整描述可视化仿真模型数据库的要求而诞生的。运用这种数据格式可以非常容易的控制模型数据，对其进行增加实时应用所需的特质。同时，OpenFlight 数据库采用树状层次化结构，可以实现两个优势：

一是可以方便地将模型按照几何特性进行有效的组织，并将其转化为能够方便地进行编辑和移动的节点的形式；二是这种树状结构非常适合实时系统进行各种遍历操作。这种树状结构由许多节点组成，每个节点有子节点或兄弟节点，通过层次和属性描述三维模型，并可以对模型面和顶点进行控制。而且这种结构的的灵活性加速了数据库的组织、模型生成、修改编辑、赋予属性和结构关系的定义，让用户轻松地组织视景数据，为超级实时图形硬件提供了优化的性能。图 3.2 所示为 MultiGen Creator 中的一个典型的 OpenFlight 模型数据库层次结构。

3.1.3 OpenFlight 模型数据库节点类型

（1）最常用的基本节点

OpenFlight 模型数据库中最基本的、最常用的节点类型是组节点、体节点、面节点和点节点四种。在层级视图中，组、体、面节点都位于默认的根节点的下方；组节点由许多群节点和对象节点构成；而对象节点又由许多面节点构成；面节点则由许多点节点构成。

1）根节点（Database Header）：根节点位于数据库层级视图的最上层，也是整个 OpenFlight 数据库树状层次结构的根。用户创建新的模型数据库时，Creator 会自动生成数据库根节点，在层级视图中根节点被显示为红色，并以“db”作为标记，用户不可以改变其名字。

2）组节点（Group Node）：组节点是一些体节点的集合，通常按照逻辑顺序把物体有条理地编排成组，可以使操作变得更加容易、快捷。在层级视图中组节点被显示为红色，默认的标志以字母“g”开头，用户可以将其改变成有意义的名字。

3）体节点（Object Node）：体节点是面节点的集合，同样地，按照逻辑顺序把组成物体的模型面有条理地组织为体，可以使数据库操作变得更方便。在层级视图中体节

点被显示为绿色，默认的标志以字母“o”开头，用户可以将其改变成有意义的名字。

4）面节点（Face Node）：面节点是一系列有序且共面的空间点，这些空间点确定的多边形就是组成模型的面。在层级视图中，面节点显示所使用的颜色跟几何模型的相应多边形所指定的颜色相同，默认的面节点标志通常以字母“p”开头，用户可以将其改变成有意义的名字。

5）点节点（Vertex Node）：点节点代表的是模型数据库中的坐标点，每个坐标点均由一组唯一的三维数据（x,y,z）来表示。由于模型数据库中通常包含非常多的顶点，所以点节点并不在层级视图中显示出来。

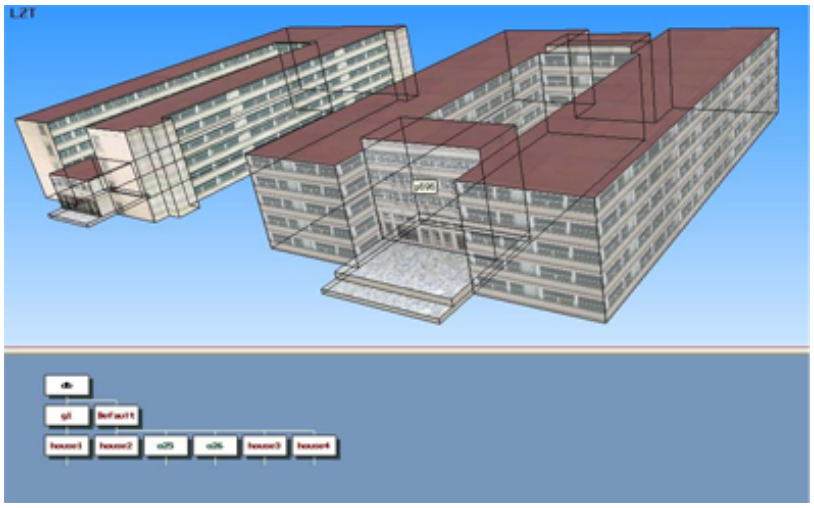


图 3.2 OpenFlight 模型层次结构视图

（2）营造特殊效果的特征节点

除了上述几种基本节点类型外，为了营造特殊的效果，用户还可以将下列特征节点加入模型数据库中，这些节点与组节点具有相同的级别。主要包括：细节层次节点（LOD Node）、自由度节点（DOF Node）、光源节点（Light Source Node）、声音节点（Sound Node）、转换节点（Switch Node）。

此外，OpenFlight 数据库还支持光点节点（Light Point Node）、外部引用节点（External Reference Node）、实例节点（Iinstance Node）、BSP 节点、裁剪节点

（Clip Node）、文字节点（Text Node）等多种类型的节点，图 3.3 所示的为在层级视图中的各种节点示意图。

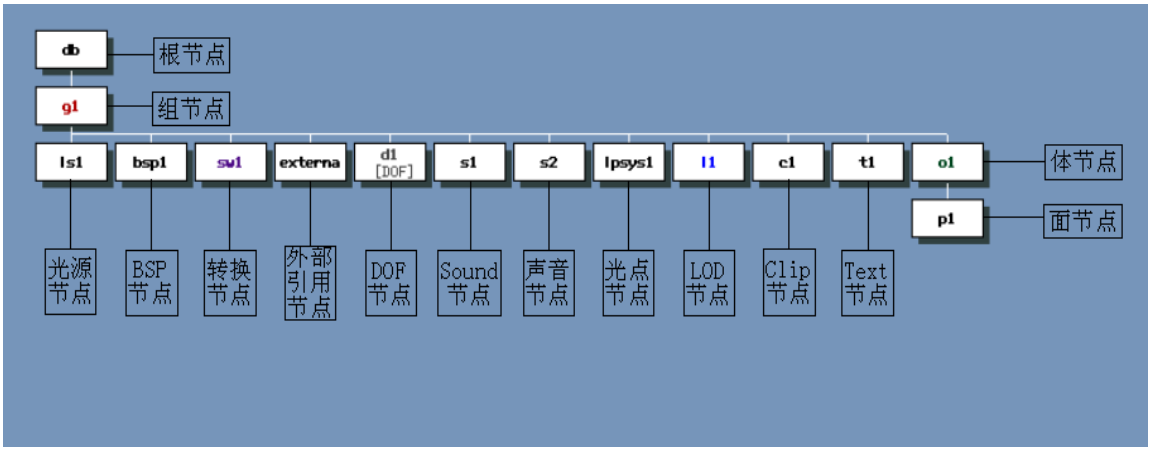


图 3.3 层级视图中的各种节点示意图

3.2 实时视景模拟驱动软件

3.2.1 Vega 简介

Vega 是MultiGen-Paradigm公司推出的先进的软件环境，是一套完整地用于开发交互式、可视化仿真应用的软件平台和工具集，它主要用于虚拟现实技术中的实时视景仿真、声音仿真以及科学计算可视化领域，最基本的功能是驱动、控制、管理虚拟场景并能够方便地实现大量特殊视觉和声音效果，支持快速复杂的视觉仿真程序，能为用户提供一种处理复杂仿真事件的便捷手段。Vega包括友好的图形环境界面LynX，完整的C语言应用程序接口API，丰富的相关实用库函数及一批可选的功能模块，将先进的模拟功能和易用工具相结合，对于复杂的应用，能够提供便捷的创建、编辑和驱动工具，其显著的特点是减少源代码的开发时间，使用户集中精力解决特殊领域内的问题而无须花费大量时间和精力去编程，从而大大地提高了工作效率。

目前，Vega 的最新版本为 3.7，又增加了许多新特征，提供了新的可选模块，并进一步提高了系统的稳定性和执行效率。同时，Vega 针对不同的用户需求，又分为多种不

同版本的产品：对于不同的应用平台，Vega 分为 Vega for NT（用于 MS Windows 平台）和 Vega for IRIX（用于 SGI IRIX 平台）；就 Vega 本身运行机制而言，Vega 又分为 VegaSP（仅支持单个处理器）和 VegaMP（支持多个处理器）；从开发和应用的角度来说，Vega 又可以分为 Vega Development（包含完整的开发包）和 Vega Runtime（仅包括运行库）两种版本。

综上所述,基于 Vega 开发实时应用的基本工作原理是,利用 ADF 文件定义各个所需要的类的实例并进行初始化设置,用户程序读入 ADF 文件中的数据,装载相关的数据库,配置图像系统,然后调用 Vega 中执行相关功能的程序库,进而渲染场景。实例一旦被创建,便被加入列表项中,所有实例通过名字被索引,当用户得到实例的句柄,就可利用 Vega 共用函数对实例的属性进行处理。

3.2.2 LynX 图形界面

LynX 用户界面非常直观，它遵循标准的窗口用户界面规则，使操作更加规范和灵活，如图 3.4 所示。它的最主要功能是定义虚拟场景中的元素属性及其相互关系，并可以实时预览参数设置的效果，最后生成用于 Vega 程序定义文件－ADF 文件。用户不需要具备专业程序员的水平就能执行 LynX 和 Vega 的应用，从而可使用户集中精力解决特殊领域内的问题而无须花费大量时间和精力去编程，节省了应用程序的开发时间。

LynX 图形环境是点击式的，用户只需利用鼠标的作、中、右键点击即可驱动图形中的对象物以及动画中的实时控制。它可以在不涉及源代码的前提下快速而容易地改变应用程序的性能，如显示通道、多 CPU 资源分配、视点、观察者、特殊效果、时间尺度、系统配置、模型和数据库等。此外，LynX 的开放性使用户可以根据自己的特殊需求赋予其新的功能。

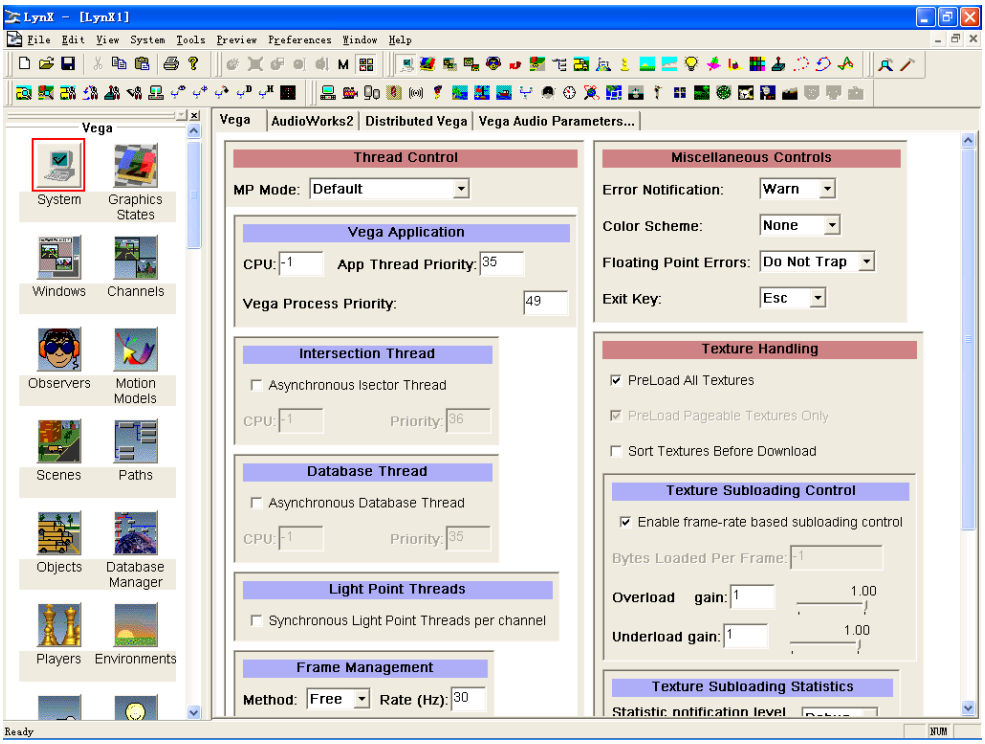


图 3.4 Lynx 用户界面

3.2.3 工程定义文件(*.ADF)数据结构

工程定义文件（*.ADF,Application Definition File）是 Vega 提供的图形界面工具 Lynx 生成的一种特殊数据格式，它对图形环境以及组成视景仿真应用的数据库组件的完全描述，其作用是方便快捷的完成初始化设置及效果预览，图 3.5 是 ADF 文件的数据结构描述。

Vega 类库中应用的通道、窗口、对象物、场景运动体、环境、图形状态等类都可以在 ADF 进行实例化操作。对每一个类都设计成节点容纳器，可容纳多个子节点。所有节点都被记录于一个节点表中。节点可以被命名，通过节点名字和遍历名字节点表我们就可以找到想操纵的节点，执行希望的操作。

图 3.5 ADF 文件数据结构

3.3 系统开发编程语言的选择

本系统开发的编程语言采用 Visual C++ 6.0。这主要是因为 Visual C++ 6.0 语言是支持面向对象技术的编程语言，是一种可视化 C++语言，汇集了 Microsoft 公司技术精华的主流产品，它最好地支持和利用了微软基本类库（MFC，Microsoft Foundation Class Library）。MFC 是微软公司为方便 Windows 程序开发而提供的一个功能强大的通用类库，它封装了大量的 Windows API。同时，Visual C++可以与 Vega 开发库函数联合编程，容易实现界面与程序的结合，从而方便、快捷地实现系统的友好界面和功能扩展编程，开发周期短，开发量较小。使用 Visual C++和 MFC 开发 32 位应用程序是目前最为广泛运用的程序设计方法之一。

3.4 图像预处理软件

图像处理软件 PhotoShop 7.0 是 Adobe 公司最新版的图像编辑软件，被广泛地应用在图像处理、绘画、多媒体界面设计等领域，可制作虚拟现实建模前期的纹理贴图，如图 3.6 所示。

图 3.6 PhotoShop 制作纹理贴图

3.5 本章小节

本章简单地介绍了楼盘 VR 系统实现的开发工具，根据研究目的，通过分析比较 VR 建模技术的理论和研究现状，获得了实现方案，最终确定了采用 MultiGen Creator 的建模和 MultiGen Vega 模型驱动与面向对象的 VC 开发平台相结合的方法实现方案。

4 开发房地产楼盘 VR 系统的关键技术

4.1 纹理映射技术

4.1.1 纹理概述

所谓纹理（texture），是指那些被映射到三维模型表面的二维图像。在构建虚拟环境或模型时，仅有体和面的几何结构是不能产生虚拟环境真实感觉的，还需要对其表面进行处理，即加表面反射和纹理。使用纹理可以在不增加多边形数量的前提下，使模型对象获得照片级的真实感视觉效果。这样做不仅提高了对象真实感，而且减少了模型的多边形，从而大大减少了场景渲染的计算量，使得实时仿真图形速度变得更快、更流畅。纹理有二维和三维图像，可以通过数码摄影或者扫描各种图片获取原始素材，然后经过适当的编辑加工获得。在 Creator3.0 中，模型纹理的映射通过 Texture 工具箱加载，如图 4.1（a）所示，模型的纹理修改通过 ModTexture 工具箱修改，如图 4.1（b）所示。

（a）Texture 工具箱　　（b）ModTexture 工具箱

图 4.1 纹理映射及修改工具箱

4.1.2 纹理映射原理

由于仿真过程中的视点会不断变化，模型对象的映射到屏幕空间的大小也会不断变化，那么，映射在模型对象表面上的纹理的实际渲染的过程中就会出现三种不同的情况：

一是一个纹理图像纹素刚好对应屏幕上的一个像素；二是多个纹理图像纹素要映射到同一个屏幕像素中；三是一个纹理图像纹素要映射到几个屏幕像素中。第一种情况属于一种理想情况，纹理图像是全尺寸映射；对于第二种情况，需要对纹理图像进行压缩过滤操作；对于第三种情况就需要对纹理图像进行被称为放大过滤的操作。当视点位置改变时，通过相应的过滤操作可以柔化、锐化或者融合纹理图像，从而可以获得更好的纹理渲染效果。如图 4.2 所示的为纹理映射原理示意图。

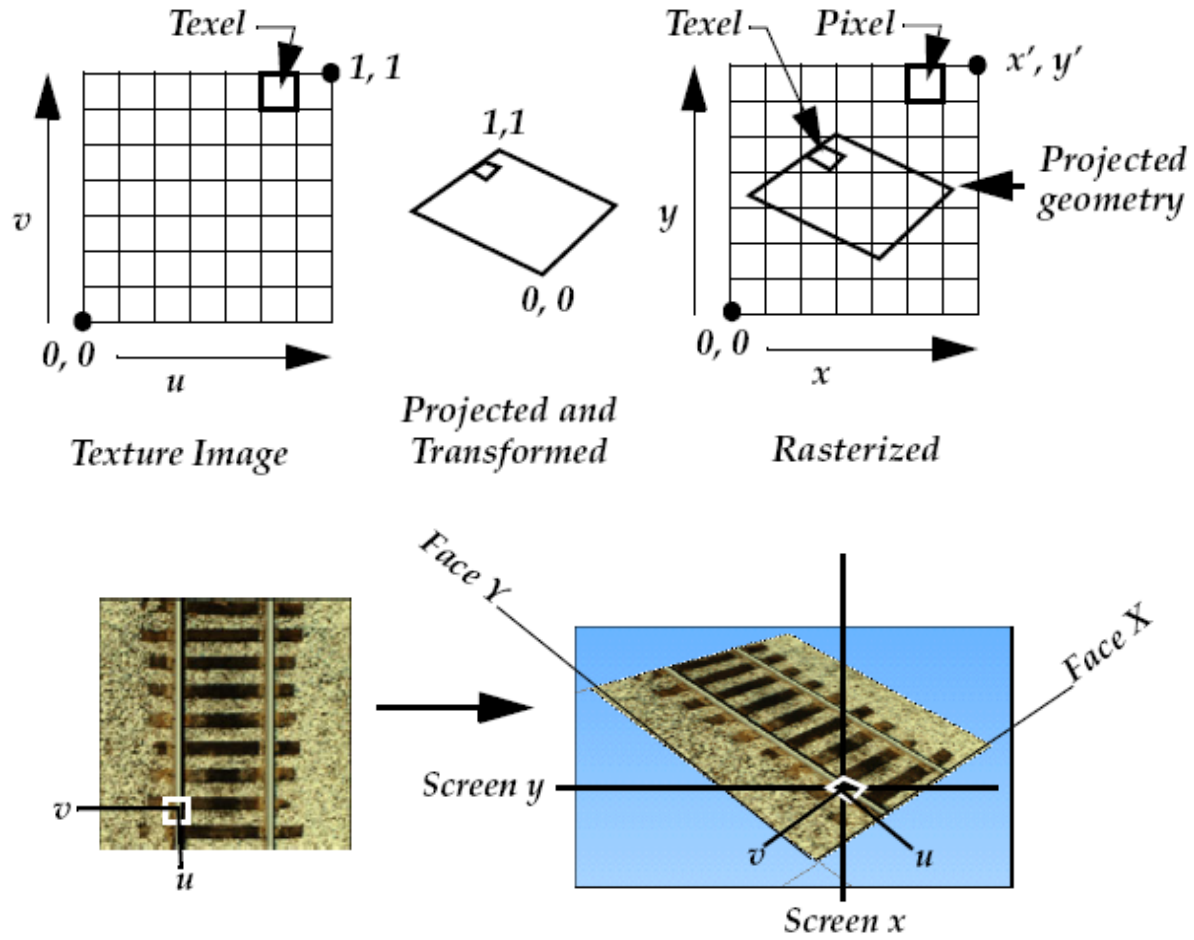


图 4.2 纹理映射原理示意图

一般来说，纹理映射（texture mapping）包括以下几个主要步骤：

（1）准备纹理图像；

（2）将纹理的 u、v 坐标映射为屏幕空间坐标；

（3）将纹理颜色与多边形颜色和材质颜色融合；

（4）进行适当的纹理过滤处理，优化纹理映射效果。

实时系统要映射模型数据库使用的纹理，必须先将相应的纹理信息（包括纹理名称、纹理存放路径等等）被加载到纹理内存中，而纹理本身并没有和几何体信息储存在一起，也就是说，纹理本省跟模型数据集是分离的。纹理内存是专门用于加载和储存纹理图像的内存空间，大多数的实时系统都有专门的纹理内存，也有的实时系统就直接将纹理图像加载到系统的内存中。为了能够更有效地利用纹理内存并保证实时系统能够正确进行纹理映射，纹理图像的两个方向上的尺寸必须是 2 的幂次方大小，比如 16×16、32×32、128×256 等等，但不要求长、宽相同。Vega 支持的纹理类型包括 RGB、RGBA、INT、INTA、JPG 等格式。纹理图像所需的纹理内存容量跟纹理图像的大小尺寸和使用的储存格式有关，具体的计算方法为：

所需纹理内存＝X 方向纹素大小×Y 方向纹素大小×颜色通道数量在 Creator 中使用纹理，首先将所需的纹理图像调入纹理调板中，指定当前纹理，然后选择目标面，使用纹理工具箱中的纹理映射工具和属性工具箱中的设置纹理工具将当前纹理映射到多边形上。

4.2 模型优化技术

4.2.1 Instance 技术

当三维复杂模型中具有多个几何形状相同但位置不同的物体时，可采用实例（Instance）技术。实例化是计算机图形学里为节省计算机的运行开销而采用的一种算法。一个实例是指对模型数据库中某个模型对象的一个参考副本，模型对象的实例跟模型的拷贝看起来是一模一样的，但实际上实例与拷贝有着本质的区别：拷贝是通过在模

型数据库中完全复制模型对象几何体创建的副本，而实例仅仅是指向模型数据库中模型对象的指针，并没有复制模型对象的几何形体。

由于通过实例化创建的模型副本并不增加模型数据库的实际多边形数量，所以，创建模型数据库的过程中适当使用模型实例，可以节省系统的内存空间和磁盘存储空间，同时还可以改善实时系统的处理性能。实例化是一个简化模型数据库的有效方法，特别是对于模型数据库中简单的、重复性的模型对象尤其如此。

创建模型对象实例的一般步骤如下：

（1）创建一个新的组节点（命名为 buildings），将其作为下面将要建立的实例节点的父节点，注意实例节点的父节点不能有其他的子节点；

（2）选择创建的组节点，点击“Parent”按钮使其作为父节点；

（3）选择目标模型对象；

（4）打开创建工具箱，单击创建实例工具按钮，进行创建实例节点。

注意，模型对象实例的默认位置与模型对象本身相同，所以，在图形视图中并不能立即看到模型对象实例，必须给实例重新定位，方法如下：

（1） 选择实例节点上的组节点（buildings），注意不要选择实例节点；

（2） 打开操作工具箱，单击移动工具按钮，弹出 Translate 对话框；

（3） 在视图中选择移动的基点，打开视图面板，选择 coord 标签，在 Current 栏的 X，Y，Z 中输入要放置实例的坐标，敲回车键；

（4） 将实例放置在目标位置后，点击 Translate 对话框的 OK 按钮，完成操作.

如果需要复制多个实例副本，同样要先选择实例节点上的组节点，然后使用复制工具箱中的简单复制工具进行复制在目标位置。如果需要把模型实例对象转变为普通模型对象，选择要转变的实例再执行“Local-DOF”菜单的“Make Geometry”。

4.2.2 使用外部引用

场景实体模型的构建，其构建方式是按照场景层次结构的划分进行的，各个层次实体景观构建完后需要进行组合集成，最终形成虚拟场景的整体模型。单独建模的实体模型要逐个添加到整体场区模型里，主要是指那些在场景中表现为静态的环境模型。还有各种其它环境景观，如：花草、树木、道路、娱乐等。通过各种技术（如复制、实例化等）组成一个整体模型。但是有些系统场景模型和构成比较复杂，文件很大，这样在进行漫游时，场景的导入和实时响应的速度就比较慢，影响了实时的效果。

在 MultiGen 中，有一种技术叫外部引用。是指在一个模型中可以设置外部参考节点，在节点下能够调用另一模型的部分或者全部，并可以重新定义被调用模型的空间位置。跟模型对象实例类似，在一个模型数据库中还对其他模型数据库进行外部引用，也相当于一个指向其他模型数据库的指针，而不需要将其中的模型对象复制粘贴至当前的模型数据库中。通过外部引用，可以有效降低模型数据库的规模，节省内存空间和储存空间，方便建模操作，提高系统资源的利用率。使用外部引用的好处是：

（1）便于场景的组织管理。对场景的构建可按其层次结构先划分若干区块，然后分别在各区块中进行构建，各小区块中的模型甚至可以单独进行建模，然后通过外部调用进行集成。

（2）可以节省内存空间，提高建模速度。在 MultiGen 中，可以通过参数设置，使得默认情况下外部引用节点不被显示。这样既节省了计算机的内存，又减少了模型的显示和刷新时间，提高了建模速度。

（3）便于模型替换。只需将外部引用的路径改变即可实现模型的替换；用外部参考调用新模型而不改变其位置，则新模型自动覆盖旧模型，方便地实现了模型的更新。

（4）便于模型的添加。通过外部引用，无需进行剪切和粘贴操作就可以将一个场

景数据库文件的整个内容加入到当前场景中。

值得注意的是：通过外部引用的模型是只读的，不能直接进行编辑，只能改变位置、方向和大小比例。如果想对被引用的模型进行修改，则必须打开原模型文件进行修改刷新。但是对于大规模的复杂场景数据库来说，利用外部参考能够极大地便利对场景模型的集成和对整个场景的组织管理。

向模型数据库中添加外部引用的一般步骤如下：

（1）在层级视图中选择适当的组节点；

（2）单击“Parent”按钮，将其作为外部引用节点的父节点，如图 4.3 所示；

（3）选择“File/External Reference”菜单命令，弹出 External Reference 对话框，如图 4.4 所示；

（4）在 External Reference 对话框的“External File Name”文本框中输入被引用的模型数据库文件名或单击文件浏览按钮进行选择；

（5）在 External Reference 对话框中，点击“Read External File”按钮，读入外部引用的模型数据库；

（6）在图形视图中，用鼠标设置外部引用的位置；

（7）单击“OK”按钮完成外部引用，或者单击“Next”按钮引用下一个模型数据库。

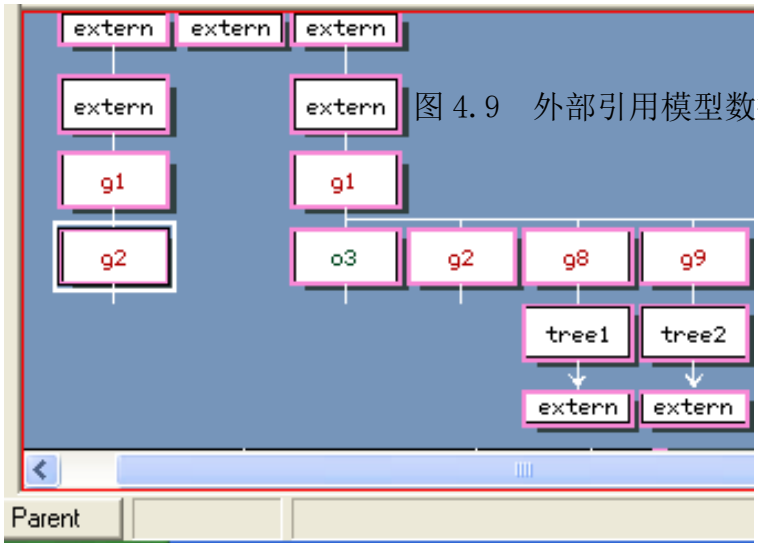


图 4.9　外部引用模型数据库

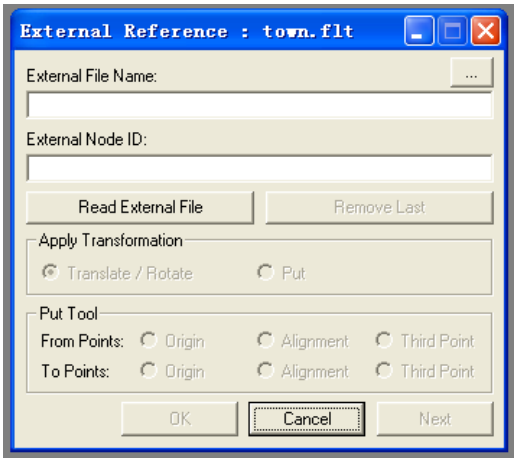


图 4.10　External Reference 对话框

注意，要想进行外部引用，在图形视图中，只能在 External Reference 模式下才能选择外部引用，选择的外部引用用红色表示，用户可以单击 Info 菜单中的 Preferences，弹出设置对话框，在“Flight”标签改变“Read External Preferences”默认设置，如果没有选择“Read External Preferences”选项，那么模型数据库中所有的外部引用都不被显示出来。

4.3 场景优化

虚拟环境氛围的营造对于虚拟场景的真实性、生动性及其对用户的感染力起着至关重要的作用。在虚拟场景中加入雾化效果可以使场景产生更强的纵深感和距离感，使远处景物产生朦胧感，从而增加了场景的真实感。

4.4 本章小节

本章对提高系统性能的三维建模关键技术（纹理映射、实例技术、外部引用）进行了简要的理论分析和探讨，重点就解决具有大规模复杂场景的虚拟现实系统“实时性”和“真实感”的矛盾问题，讨论了综合应用各种建模关键技术，以结构优化和模型优化分类方式，提出了基于 MultiGen Creator 的三维虚拟场景建模综合优化技术和方法。这些技术和方法的介绍和分析为系统的研究与开发提供了理论基础保证。

5　房地产楼盘 VR 系统的实现

5.1 系统分析与总体设计

5.1.1 系统分析

要实现房地产楼盘系统交互式漫游，需要构建许多的三维场景。本文依据我校临潼校区建筑规模，仿真校区布置、环境、空间关系，使参与者可以走进校区，身临其境地感受校区环境，不用亲自去实地就可以了解校区情况。可通过两个方面来实现整个过程：第一，建模技术是建立房地产楼盘 VR 系统的基础，因此首先要根据校区实际建筑布局

及环境对各个对象进行建模，通过对这些单个对象模型的构建，然后进行模型对象组合放置，构建出整个系统的基本框架。第二，运用程序进行交互能力的控制，要进行楼盘演示，达到实时交互效果，我们必须编写具有可操作性的交互程序。

5.1.2 系统总体设计

在系统的最初开发阶段，首先要对所虚拟的系统进行总体分析和总体设计，从而为后续工作做好准备，所以，总体设计的确定是系统开发过程中至关重要的一环，它的好坏直接影响着系统整体功能的实现以及后续开发工作的可行性和难易程度，需要对其进行全面而细致的设计和反复推敲。校区重要是由建筑物和绿化组成的，因此在进行构建虚拟场景的时候首先必须对熟悉建筑物的位置及尺寸等。如果这些资料和空间关系不是很熟悉和完备，那么在将单个模型进行系统组合的时候，就会可能因为尺寸及空间的关系而无法准确地结合，不但费时费工，而且影响仿真的真实感。同时，有准确的设计尺寸，在进行纹理贴图制作时，也可以按照相应尺寸和规格进行制作，从而方便了建模过程。

5.2 系统的模块结构和模块功能

5.2.1 模块结构的优化设计

实现房地产楼盘虚拟现实系统的核心是模块的功能实现和结构优化，所以在开发的前期，必须要对系统模块进行认真地分析组织，为更好地实现各模块功能奠定基础。针对房地产楼盘系统的结构和特点，作者采用树状层级结构方式进行优化模块结构，使得各个模块按照所要实现的功能特性进行有效的组织。如图 5.1 所示。

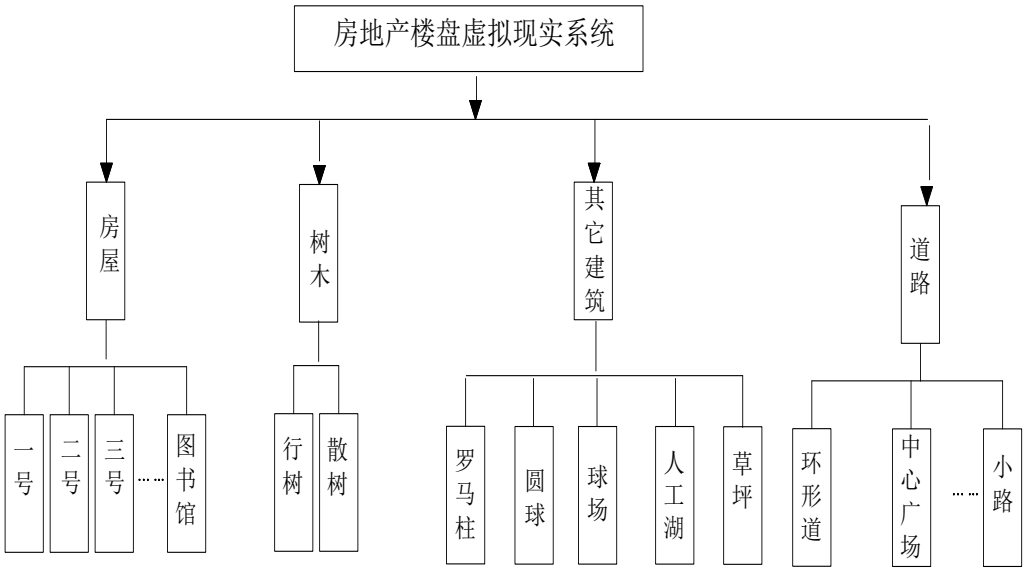


图 5.1 虚拟现实系统模块结构

5.2.2 模块的功能实现

通过建立各个模块，完善模块的功能，可以直观地模拟房地产楼盘整个系统，从而在个人计算机或者工作站实现方便获取楼盘、小区的信息，找到适合自己的单元。

5.3 系统的实现过程

5.3.1 收集资料数据

建造模型的首要工作是收集模型的数据资料，在进行建模前，必须要确定系统场景的实体几何尺寸，相互位置坐标等。如已有要建模区域地形图就更方便了，本文是将我校临潼校区CAD格式的地形图转换成dxf格式导入Creator建模界面，在平面图上进行建模。

5.3.2 制作纹理贴图

在房地产楼盘虚拟现实系统中，由于模型精细程度的限制，场景的真实感很大程度上要靠纹理来体现。由于直接在 Creator 中不能对纹理图片进行全面细致的处理，要借

助其他系列图像处理软件来完成。素材收集时用数码相机拍的实景照片，借助于Photoshop 图像处理软件进行纠正处理。

由于在 Creator 中灯光的功能不够强大，所以在制作贴图的时候有必要在Photoshop 中就控制其贴图的灯光，并按照实际情况来制作贴图，来控制其比例大小，以免在 Creator 中出现拉伸等现象。贴图的大小切记是 2 的 N 次方，这个贴图的大小很重要，如果不是 2 的 N 次方，则在转 FST 的时候会出现错误，在 Vega 中也会出现错误，有时候导致图像直接不出来，或者是图像会堆成一堆等等的情况。然后利用插件存储为 RGB 或 RGBA 格式，作为模型纹理库，供 Creator 调用。像地面树木这种情况都要使用像 RGBA 这样子的贴图，最好是透明的贴图，除树之外，别的好多地方的帖图也要是透明的。

在贴图的时候一般选用选择 Creator 三点贴图方式，如果出现圆面等情况，可以使用 U,V 属性来调整它。实际情况按实际情况对待，适当的时候可以选择别的贴图方式，例如球形贴图、表面贴图和环境贴图等。

值得注意的是：作好的纹理贴图一般要放在相同的路径下，这样做方便管理，即使换了工作站也没有关系，可以把它转换为相对路径，在 Creator 的纹理调板里面的 Info 菜单里面单击 List Textures 打开对话框，在 Change All Paths 里面找到它进行修改。

5.3.3 数据格式转化

在实现房地产楼盘 VR 系统的过程中，经常需要进行数据格式的转化，这就需要根据自己的数据格式编写数据转化程序，对于非专业的程序员或者其它专业的应用者来讲，如果没有编写数据格式的技能，则在制作纹理、进行建模及视景驱动过程中遇到一些麻烦，以下将介绍几种通过插件的数据转化的方式。

（1）RGB 格式转化。在用 Photoshop 制作纹理贴图的时候，纹理的存储方式需要小心处理，一般将其大小保存为 2 的 N 次方，格式保存为 RGB 或 RGBA 格式。而 Photoshop 没有自带的 RGB 格式转化功能，我们这时可以借助 Creator2.6 或者 Creator3.0 自带的一个插件进行转化，转化方法步骤如下：

① 找到 Creator2.6 或者 Creator3.0 安装盘里的 RGBFormat.8BI 文件，一般存放路径为：Creator3.0\Free_Stuff\Photoshop_Plugins；

② 选择 RGBFormat.8BI 文件，复制此文件；

③ 打开安装目录，将 RGBFormat.8BI 文件粘贴在 photoshop\Plug-Ins\File Formats 文件里；

④ 重新启动 photoshop，新建一个纹理图，单击菜单 File/Save as，弹出 Save as 对话框，在 Format 项目下拉条目选择 SGI RGB （*.RGB；*.INT），进行保存。

（2）FLT 格式转化为 FST 格式。用 Creator 创建好的 FLT 格式模型文件，在换别的机子或更改了纹理目录等后，模型的纹理效果就会丢失，很不容易进行管理。所以一般需要把 FLT 数据格式的模型转化成 FST 格式的二进制文件，并将模型纹理包含其中，这样做不仅可以大大加快系统的加载速度，提高运行效率，特别是对于大型的模型数据库而言特别实用，而且 FST 格式的模型由于具有不可逆性，所以从另一个方面看，又同时具备了保护自己模型的知识产权的目的，Creator 中不能打开 FST 格式的模型的，并且将纹理包含后，FST 格式文件成为了一个单独的文件，不再受到路径的困扰，发布仿真应用程序时也不用再附带纹理文件夹，简化了操作，增加了其运用的灵活性。 转换的方法有两种：

方法一：

① 创建一个新的 ADF 文件；

② 将要转换的 FLT 模型依次添加到 Objects 面板中；

③ 保存该 ADF 文件，假设命名为 Moxing.adf；

④ 打开 DOS 窗口，转到保存 ADF 文件的目录下；

⑤ 在命令行提示下，键入如下命令：

>objconvert -A Moxing.adf -s fst -i

-A 表示转化指定 ADF 文件中的所有模型对象

-s fst 表示指定转化后的模型格式为 fst

-i 表示将纹理包含在 fst 文件中

⑥ 回车，转化完成！

方法二：

如果自己有一个名为 Mulde 的网友做的 fstexport.dll 插件的话，可以直接在 Creator 中完成转换工作，具体方法如下：

① 正常安装 Creator 和 Vega 软件，本插件在 Creator 2.6 以上版本测试通过；

② 假设 Creator 安装在 D 盘，解压后，把 fstexport.dll 放到安装目录 D:\Program Files\MultiGen-Paradigm\config\creator\plugins 下即可；

③ 启动 Creator 后打开要转换的 flt 文件；

④ 选择菜单 File->Write FST File ...，如图 5.1 所示；

⑤ 在弹出的对话框中输入存放文件路径和文件名称，或者直接选择浏览按钮进行选择路径，如果需要纹理，选择包含纹理于 FST 文件中，单击输出按钮.

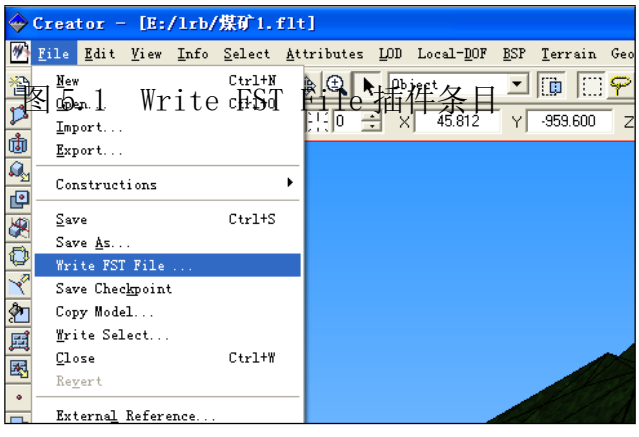


图 5.1 Write FST File 插件条目

5.3.4 建立实体模型

房地产楼盘虚拟现实系统场景模型是整个实时视景仿真的核心，模型的好坏，直接影响到运行的效果和场景的逼真度。根据楼盘系统建模的需求分析，不同模型有着不同的重要性，同时当前的模型也有着不同的建模要求，因此在建模的过程中，对不同的模型给予区别对待。一般来说，对于越重要、越需要突出显示的模型，模型就建得越详细，面片数也越多；对于越不重要的模型，模型就建的越简单，面片数也就越少。这样就能充分利用限有的宝贵内存和CPU进行实时交互。几何模型及场景建完后，给模型的表面加上各种材质，设定各种光照条件，进行几何表面纹理贴图，接着要把初步模型加入虚拟现实系统进行验证，不断修改使得模型看上去更加光滑、真实、有质感，使得模型更加符合实际要求，成立最终模型，生成(*.flt)格式模型文件。

5.3.5 系统场景合成

运用 Vega 的 Lynx 功能提供图形用户界面，创建用于实时应用的 ADF(Application Definition Files)文件，在图形用户界面下，修改参数，最终生成（*.ADF）格式文件，为开发出强大的实时应用程序作准备。

Vega 的 Lynx 功能提供了一个图形用户界面，用来创建用于实时应用的 ADF(Application Definition Files)文件，在图形用户界面下，可以容易的修改参数，使得非专业编程人员能开发出强大的实时应用程序。ADF 描述了用于实时应用的 OpenFlight 文件、运动体及路径的特殊效果、环境效果及其他功能。

5.4 Vega 应用程序的建立

5.4.1 Vega 应用程序的主循环

在 ADF 文件之外，通过编写程序来实现房地产虚拟现实系统的实时性和交互性需求。

Vega 编程类似于 C 编程，实际接口是 C，包括完整的 C 语言应用程序接口，为软件开发人员提供最大程度的软件控制和灵活性。对于 Windows NT 平台上的 Vega 应用，主要有三种类型：控制台程序、传统的 Windows 应用程序和基于 MFC(Microsoft Foundation Classes)的应用。但无论是哪一种应用，建立 Vega 应用程序都要分为 3 个步骤：

(1)初始化：调用 vgInitSys 函数初始化系统并创建共享的内存区和信号区。

(2)定义：通过创建需要的事件和需要的类来定义系统。

(3)系统定义：进行系统配置，使 Lynx 下的 ADF 文件中的定义与 Vega API 函数调用结合起来，最后调用 vgConfigSys 函数完成定义。

Vega 应用程序包括 vgSyncFrame 和 vgFrame 函数的调用，通常由每个主循环或者每次需要一个新的显示时调用这些函数，详细过程如流程图 5.7 所示。

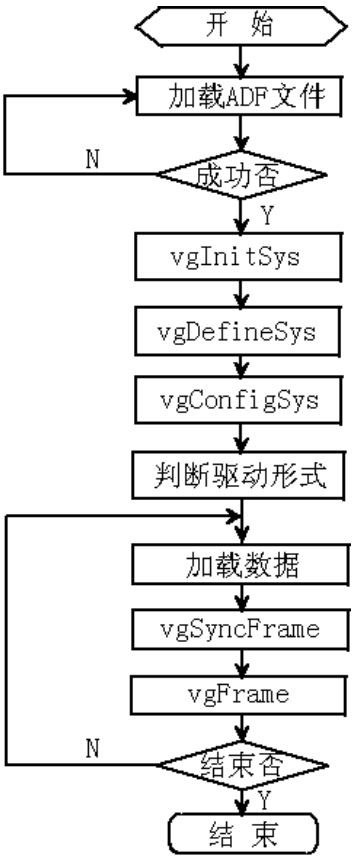


图 5.7 程序流程图

5.5 本章小节

本章对系统进行了总统的分析和设计，采用树状层级结构方式进行优化模块结构，介绍了楼盘 VR 系统的实现技术路线和方法，包括纹理制作、数据格式转化方式、建立实体模型及系统场景合成等技术，并介绍了 Vega 应用程序建立的实现方式。

6 结论与展望

随着计算机硬件和软件的发展，在个人计算机上或实验室工作站上发展虚拟现实技术已不再成为梦想。VR 是计算机技术、VR 技术、图形图像显示技术等诸多高新技术的综合运用。房地产楼盘虚拟现实系统的研究与应用，无疑改变了传统的样板房的售楼方式，对开发商的楼盘销售、消费者购房及房地产市场的动作等各方面都会有很大的影响。本文初步研究了实现房地产楼盘 VR 系统的思路和解决可控可视化的技术问题的方法。总结理论研究和开发实践，论文主要介绍完成了：虚拟现实的相关概念、虚拟现实的三维图形几何变换技术与立体显示技术；三维建模关键技术（纹理映射、细节层次、实例技术）；房地产楼盘 VR 系统的实现技术路线和方法，包括纹理制作、数据格式转化方式、建立实体模型及系统场景合成等技术，对系统进行了总体的分析和设计，采用树状层级结构方式进行优化模块结构，介绍了 Vega 应用程序建立的实现方式。

本文以一个虚拟校园为例，借助于建模软件 Greator 和场景驱动软件 Vega，介绍了虚拟楼盘实时漫游的实际开发应用，详细介绍了虚拟场景模型创建，简要介绍了 Vega 应用程序的执行过程和漫游方式实现。并给出了漫游效果图。但由于时间和能力的限制，

未能对系统设计所包含的所用模块进行全面的研究与开发，实现功能比较单一，还需要进一步的学习和研究，开发更多的特殊效果。

致 谢

本论文的研究工作是在导师赵国梁老师的细心关怀和指导下完成的，从论文选题到撰写完成的整个过程，无处不倾注着导师辛勤指导的心血和汗水。在即将毕业之际，向我的导师致以最诚挚的谢意！

同时衷心地感谢在这几年的学习中给予我关怀和指导的所有老师和同学！是他们使我学到了许多宝贵的知识并拥有了不懈的动力。

感谢测量系的领导和所有老师给我们提供了良好的电脑设备，使我们顺利完成了毕业设计。

最后，感谢各位评审专家在百忙之中抽出宝贵时间对我的论文进行审阅！

衷心谢谢各位评委的赐教和指正！

参考文献

[1] 黄铁军.虚拟现实导引[J].计算机世界,1997.

[2] 汪成为.灵境是建立人机和谐仿真系统的关键技术[J].系统仿真学报,1995.

[3] 王守清.计算机辅助建筑工程项目管理[M].北京:清华大学出版社,1996

[4] 张秀山等.虚拟现实技术及编程技巧[M].长沙:国防科技大学出版社,1999

[5] 彭华林,欧青立,李仁发.虚拟现实技术在电子设计与测试中的应用[J].来自湘潭矿业学院学报,1998.

[6] 彭华林,欧青立,黄丹.电子测试虚拟化实践教学的新模式[J].实验室研究与探索,1998.

[7] 王乘,周均清,李利军.Creator 可视化仿真建模技术[TP].武昌：华中科技大学出版社,2005.

[8] 王乘,李利军,周均清,陈大炜.Vega实时三维视景仿真技术[TP].武昌：华中科技大学出版社,2005.

[9] 王红兵.虚拟现实技术—回顾与展望.计算机工程应用,2001,048－51.

[10] 段新昱.虚拟现实基础与VRML编程[M].高等教育出版社，2004.

[11] 龚卓荣.Lynx图形界面[M].国防工业出版社,2002.

[12] 胡小强.虚拟现实技术[M].北京邮电大学出版社,2005.

[13] 赵伟. 虚拟现实中LOD技术的研究. 山东科技大学硕士学位论文，2003，3～5.

[14] 杜健. MFC框架下基于Vega的航海仿真系统视景驱动程序的开发. 大连海事大学硕士学位论文，2004，10～11.

[15] MultiGen Creator Desktop Tutor Version 3.0[Z].MutiGen-Paradigm,Inc,2004.

[16] 翟丽平. 基于 MultiGen 的虚拟现实三维建模技术研究与实现. 重庆大学硕士学位论文，2005，40－41.

[17] 张晋堂. 基于 Vega 的视景仿真技术在无人机舱中的应用研究. 哈尔滨工业大学硕士学位论文，2005，20－21.

[18] 赵伟. 虚拟现实中 LOD 技术的研究. 山东科技大学硕士学位论文，2003，3－5.

[19] 胡小强.虚拟现实技术[M].北京邮电大学出版社，2005.

[20] 洪炳镕,蔡则苏等.虚拟现实技术与应用[M].国防工业出版社，2005.

[21] 汪成为等. 灵境（虚拟现实）技术的理论、实现及应用[M]. 清华大学出版社，1996，5～6.

[22] 吴启迪.系统仿真与虚拟现实[M].化学工业出版社，2002.

[23] 石教英.虚拟现实基础及其应用[M].科学出版社，2002.

[24] 龚卓荣.Vega 程序设计[M].国防工业出版社,2002.

[25] 龚卓荣.可选模块的使用与开发[M].国防工业出版社,2003

目　录