教育部高等学校广播影视类专业教学指导委员会“十一五”规划教材

总主编　王建国　孙立军

Maya特效制作

曾庆珉 张耀华　编著

上海交通大學出版社

教育部高等学校广播影视类专业教学指导委员会“十一五”规划教材

Maya特效制作

Maya Texiao Zhizuo

曾庆珉　张耀华　编著

上海交通大学出版社

图书在版编目（CIP）数据

Maya特效制作/曾庆珉，张耀华编著. —上海：上海交通大学出版社，2009
教育部高等学校广播影视类专业教学指导委员会“十一五”规划教材
ISBN 978-7-313-05540-8

Ⅰ.M… Ⅱ.①曾…②张… Ⅲ.三维—动画—图形软件，Maya—高等学校：技术学校—教材 Ⅳ.TP391.41

中国版本图书馆CIP数据核字(2008)第202199号

Maya特效制作

曾庆珉 张耀华 编著

上海交通大学出版社出版发行

（上海市番禺路951号 邮政编码：200030）

电话：64071208 出版人：韩建民

上海锦佳装璜印刷发展公司印刷 全国新华书店经销

开本：787mm×1092mm 1/16 印张：7.5 字数：199千字

2009年1月第1版 2009年1月第1次印刷

印数：1～3550

ISBN 978-7-313-05540-8/TP·711 定价:38.00元

版权所有 侵权必究

教育部高等学校广播影视类专业教学指导委员会
“十一五”规划教材编审委员会

顾问名单

金德龙　国家广播电影电视总局副总编辑、宣传管理司司长
余培侠　中央电视台青少节目中心主任、中国动画学会会长
张松林　中国动画学会原副会长、秘书长
贡建英　中国动画学会副会长兼秘书长
曲建方　国际动画协会会员、中国电视艺术家协会卡通艺术委员会副主任
曹小卉　北京电影学院动画学院原副院长
蔡志军　中央电视台动画创作部主任
赵　欣　中央电视台动画创作部制片

成员名单

王建国　陈　龙　陈信凌　毕一鸣　布和温都苏
董广安　高晓虹　蒋贻杰　梁小庆　刘民朝
王诗文　谢晓晶　张瑞麟　郭卫东　孙立军
李　霞　覃晓燕

序

PROLOG

21世纪，人类社会进入了信息时代与知识经济时代。在这个飞速发展的时代里，经济全球化与文化多元化已经成为不可阻挡的历史潮流。随之而来的是跨文化传播在全球的迅速兴起，而影视艺术作为当今世界影响力最大的艺术创造和文化传播方式之一，在跨文化传播中具有最广泛的观众群和覆盖面。

随着广播影视事业在全国的迅速发展和产业属性的显现，对广播影视人才的需求也越来越大，近年来，我国广播影视类专业高等教育取得了长足的发展，为广播影视系统输送了大量的人才。随着广播影视行业的迅猛发展，社会对广播影视类人才提出了更高的要求。进一步深化人才培养模式、课程体系和教学内容的改革，提高办学质量，培养更多的适应新世纪需要的具有创新能力的广播影视高素质人才，是广播影视教育的当务之急。

作为广播影视教育的重要环节，教材建设肩负着重要的使命，新的形势要求教材建设适应新的教学要求。本教材应针对高等学校学生自身特点，按照国家高等教育的特点和人才培养目标，以素质教育、创新教育为基础，以学生能力培养、技能实训为本位，使职业资格认证培训内容和教材内容有机衔接，全面构建适应21世纪人才培养需求的高等学校广播影视类专业教材体系。广播影视类专业教学指导委员会组织编写的“十一五”规划教材，主要包括影视动画、影视广告、新闻采编与制作、主持与播音、电视节目制作、摄影摄像技术等专业系列教材，本系列教材的出版，必将对高等学校广播影视类专业的人才培养和教育教学改革工作起到积极的推动作用。

本系列教材的出版，得到了教育部高等教育司领导、国家广播电影电视总局人事教育司领导及行业专家的大力支持，得到了国内众多同类院校的大力协助，在此对他们表示衷心的感谢！同时，我们也希望广大师生和读者给我们提出宝贵意见，使教材更加完善。

高等学校广播影视类专业“十一五”规划教材编写委员会

王建国 教授

此为试读，需要完整PDF请访问：www.ertongbook.com

内容简介

本书利用Maya制作数字特效为入门指导，以动力学相关功能模块的应用为核心，通过对关键功能设置等的说明，结合样例进行分析与引导，力图帮助读者理解特效制作中的关键模式。本书概要性地介绍Maya的节点体系和MEL、表达式编程的应用，逐一讲解Maya主要动力学模块的核心内容，示范在特效表现中的运用。学习本书之前，读者需拥有一定制作完整三维动画作品的经验，并掌握常用的后期合成与特效技术。

课程与课时安排

章 节	内 容	课 时	理论教学	课内实训
第一章	数字视觉特效与Maya	12	7	5
第二章	刚体	10	5	5
第三章	粒子特效	16	8	8
第四章	柔体与布料	12	6	6
第五章	流体	10	5	5
第六章	综合实例——异时空来客	12	6	6

目录

CONTENTS

第一章　数字视觉特效与Maya

数字视觉特效越来越频繁地被运用在影视广告等视觉产品中，现今对于许多观众已不再是陌生的技术名词了。Maya是一款优秀的制作三维动画的软件包，除了常规的建模、动画、材质灯光和渲染功能，它也包括了多种特效工具。本章简要介绍了当前数字特效的常用技术，并对Maya的特效制作、体系结构与编程开发等概况逐一进行说明。

主要内容：介绍数字特效的常用技术和应用，Maya的节点体系结构和功能模块，示范和讲解MEL及表达式编程的基础知识。

学习重点：Maya的节点和特效模块，MEL和表达式编程基础。

第一节　数字视觉特效

在电影和电视这些连续影像的媒体诞生的早期，就已经出现了多种视觉特效技术，例如倒播、慢动作、叠画、微缩模型等。但由于胶片和模拟式电子信号处理的种种限制，特效表现的品质和自由度都有很大限制，几十年之间没有多少突出的创新。而随着电子计算机的大众化，不断提升性能，缩小尺寸和下降成本，越来越丰富的软件和周边设备利用计算机高自由度和高质量的计算处理能力，影视特效领域出现了飞跃式的突破和发展。

数字视觉特效技术五花八门、日新月异，即使仅是Maya提供的工具及其运用组合就非常丰富，本书不可能做到一一涉猎，只能通过一些常见特效的样例，引导学习者理解特效的设计制作思路和解决问题的途径。

三维动画和数字特效从表现效果上来分，一个方向是模拟真实、尽量不露痕迹，另一个方向是艺术性的、抽象性的视觉表现。相对而言，由于长期生活的经验，人类视觉拥有敏锐的观察力或者说非常挑剔，所以，要求三维动画和数字特效拟真的技术难度相当高，制作工作量也很大。

如果从画面组成的元素来说，数字视觉特效则有纯粹计算机生成的“虚”和拍摄现实素材的“实”。对于特效制作的整个过程来说，将各种虚实元素进行处理和合成的视频后期制作阶段，是不可或缺的。利用Maya制作的视觉元素有明显的三维空间特征，但Maya不涉及后期合成，因此本书将不会讨论后期特效与合成方面的内容。但是理解和掌握数字后期主要技术，对于三维特效的设计和制作来说是非常必要的前提。

常用的后期处理是指这几个方面：① 单一视频片段的剪辑和变速。② 多层画面合成，包括遮挡、叠加、光色加深等。③ 两段画面过渡，如由简单推移切换到复杂的过场特效。④ 色彩处理，包括多种调色、阀值、反相等特效处理。⑤ 深度通道（三维动画渲染）运用，如景深模糊、雾效。⑥ 风格化处理，包括浮雕、艺术笔刷、布纹、模糊、锐化、去噪

等。⑦ 素材合成前处理，通常指蓝、绿屏抠像。⑧画面变形，包括变换、球形扭曲、网格式自由变形、变形融合等。⑨ 二维粒子系统，如火、烟雾、星光等效果。

比如，如果想把利用计算机生成的恐龙或者机器人放到实拍的城市街道或者野外的画面中，那就需要把虚实元素的空间透视、光影和随时间的变化搭配得宜。在Maya中，功能模块Live（如图1—1所示）就可以完成实拍素材中物体和摄像机的三维虚拟定位。

本书专注于三维动画中的特效表现。“特效”并不具备严谨的、长期不变的定义。在三维动画制作领域，我们一般把通过常规的建模、着色、关键帧动画等直观的方法可以制作出来的内容，排除在“特效”定义之外。换句话说，那些用常规方法制作有相当大困难、或者工作量过大的视觉效果，即为特效技术需要解决的问题。尤其是在动画表现方面，像漫天飘舞的雪花、风中的草地、飘摆的衣裙、跳动的火焰、翻滚的烟雾等，若利用变形和关键帧动画等方法来制作，不仅是模拟现实中那种复杂多变的细节非常困难，而且又加大编辑、修改的工作量。

为解决这类大规模复杂动画的表现问题，人们开发了多种动画自动生成工具。这些工具生成动画的计算，常是基于简化的物理规则，因此也被称为动力学（Dynamic）工具。计算机辅助生成的这些特效动画，相对于手工动画而言，效率不知要高多少倍。

常见三维特效技术及应用主要用于以下几个方

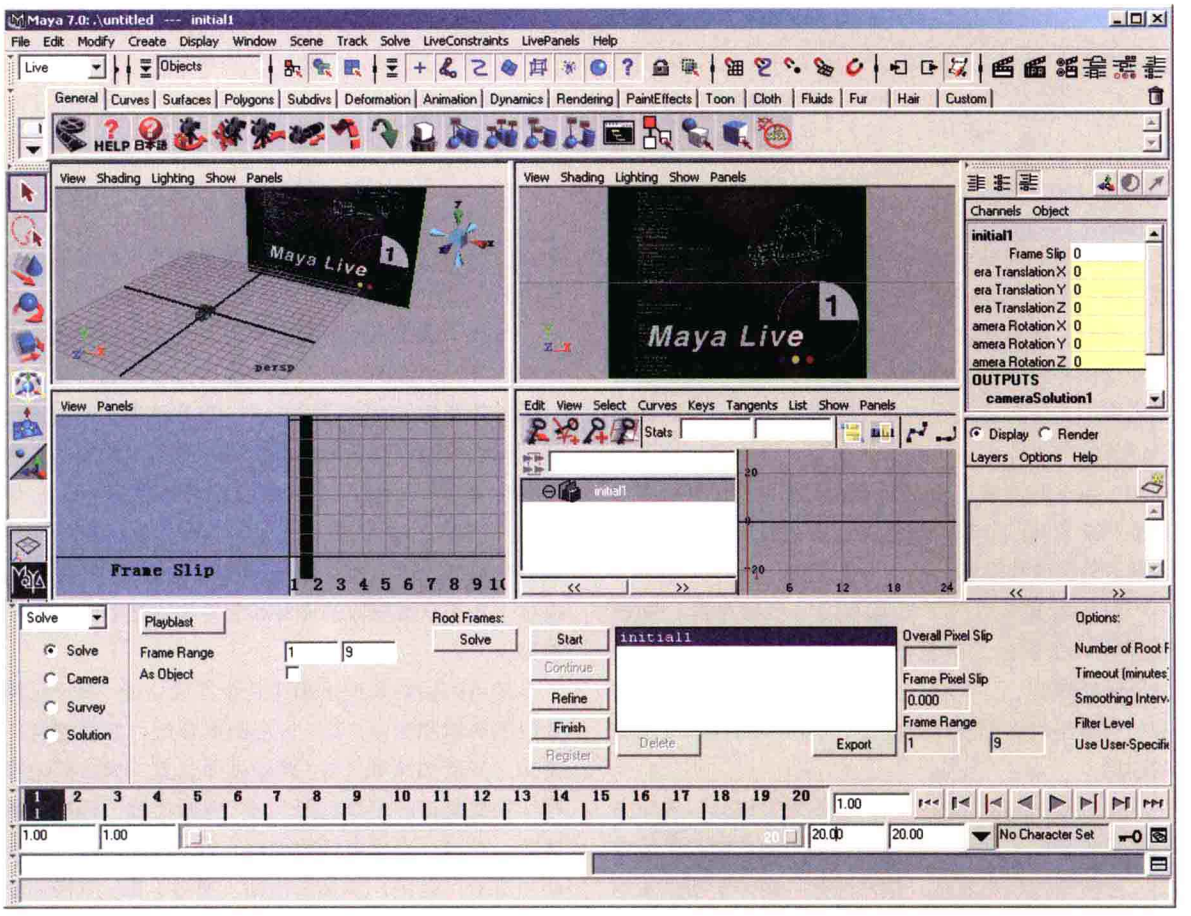

图1—1

面：① 粒子系统，用于表现火、烟、云、流星、爆炸、水花、集群动物或人物等，是最常用的三维特效。 ② 刚体，用于表现物体碰撞、破碎、倒塌等复杂动画过程。③ 柔体，用于表现枕头、果冻、肌肉等柔软弹性物体的变形动画。④ 布料，用于表现衣服等纺织品的摆动、褶皱动画。⑤ 头发，用于表现发型的设计和自然摆动。⑥ 车体，用于表现动力学组合，模拟驾驶时汽车车身与车轮的运动表现，常见于实时游戏。⑦ 人偶，用于表现动力学组合，模拟失去自主动作能力后的人形角色之跌倒、滚动等运动。⑧ 群体动画，利用AI技术产生大量动作各异的不同角色。⑨ 流体，用于表现细腻逼真的水流、水花、云烟、爆炸等，由于计算量巨大，处理时间长，目前仅运用于影视动画中。⑩ L-System，运用L-System生成植物等自然景物的模型，可制作生长动画。

通常许多动力学特效的计算机生成过程，需要花费相当长的时间。在三维互动游戏和军事模拟等实时图形应用中，受到每秒20次以上更新计算的效果要求限制，所以不得不精简特效表现而致使许多逼真的细节缺乏。设计和制作特效时需要仔细考虑技巧性地利用纹理贴图、后期处理等方法，以尽量小的代价制作。

特效工具的运用和控制经常涉及到一些物理意义的参数，还可能需要数学公式的帮助，跟关键帧动画等常规方法比起来，就相对复杂一些。由于设置参数经常较多，又需要使用者有一定的数学和物理知识，运用理性思维解决问题，因此特效制作是属于偏向技术型的工作。

包括Maya在内的许多软件，提供了一些精简的特效工具或者效果组合模板，方便于技术能力不足的三维动画设计师运用。当然便利也意味着局限，这些特效的表现空间较为狭窄。当然，对于只需要制作一些简单特效进行装饰的用户来说，只要阅读软件附带的功能说明文档、参考一些简单样例即可运用。

学习特效时，不能依赖重复各种样例的制作过程。这样限于死记参数设置和流程操作的方法，是难以应付实际应用中千变万化的特效需求的。在对每一样例的学习中，应仔细理解每一主要步骤、相关工具或操作的目的，理解核心参数的几何或者物理意义，学会将问题的处理过程进行分解和组织。先通过简洁的练习来强化对各技术单元的理解和控制，再来尝试较复杂的综合性制作。对于任何技术性较强的领域，由归纳到演绎的思维过程，都是学习方法的核心。

数字特效不仅涉及许多技术和理性思维，而且要想制作令人瞩目的视觉效果，还需要在艺术表现和创造性上投入很大的精力。本书内容虽未涉及特效的艺术设计，但仍然建议读者在鉴赏一些优秀数字特效作品的过程中，除了学习他们的技术技巧和制作流程，还要观摩其作品中的艺术设计和创新表现。

第二节 Maya的架构与特效

一、用户界面

学习本书的读者，对Maya的界面（如图1-2所示），常用功能操作和一般三维动画制作方法应已较为熟悉，在此仅简单介绍。

如果按功能模块切换，Maya的菜单模块分为Animation（动画）、Modeling（模型，8.0版本之后拆分为Polygons和Surfaces两个模块）、Dynamics（动力学）、Rendering（渲染）、Cloth（布料，8.5版本之后改为nCloth）和Live（实景匹配）。

按其类型与功能特色还可以划分出Subdiv Surfaces（细分表面）、Paint Effects（画笔）、Soft/Rigid

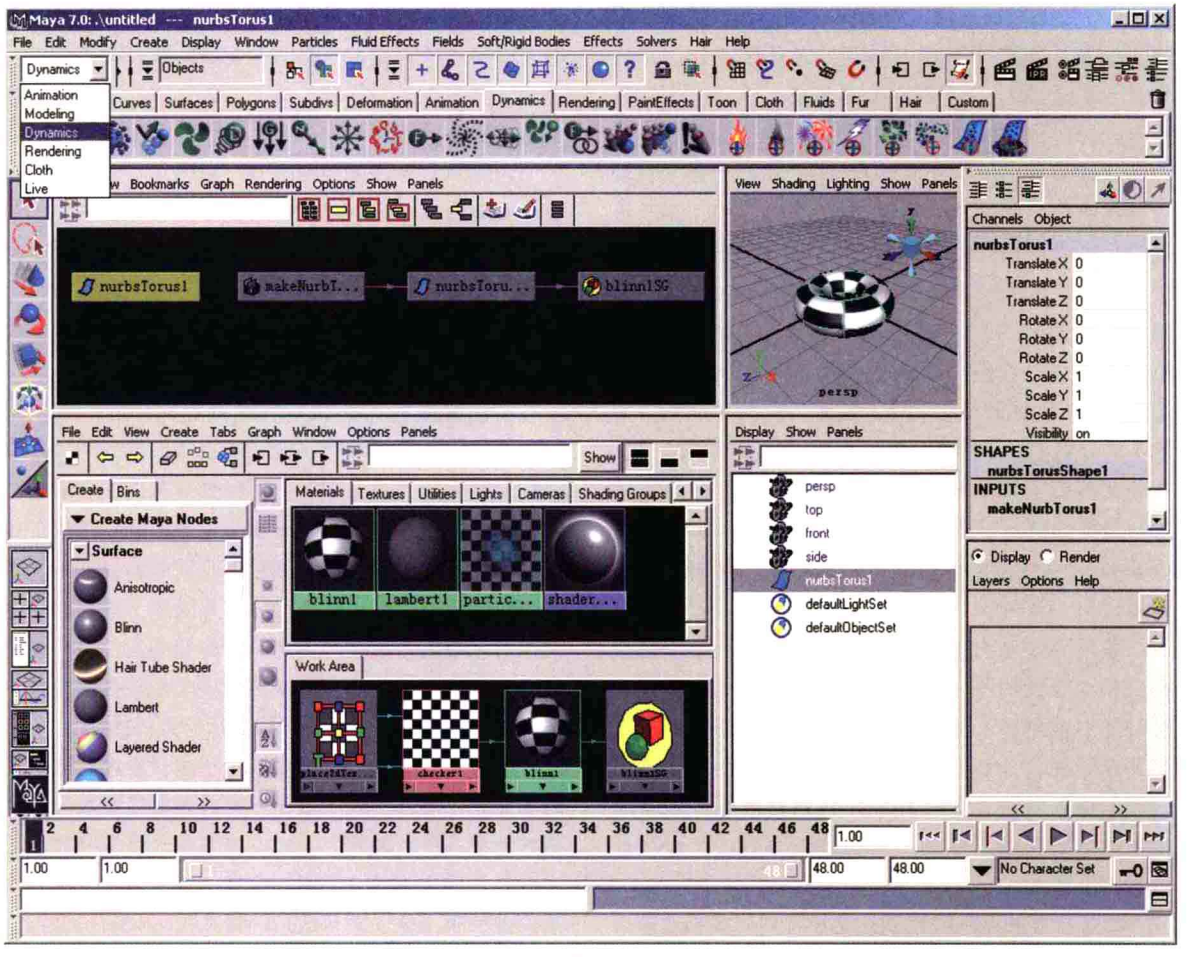

图1-2

Bodies（柔体与刚体）、Particles（粒子系统）、Fur（毛）、Hair（发）、Fluid Effects（流体）、Toon（卡通）、Muscle（肌肉，2008版本）等模块。

Maya的主要界面元素包括主菜单、主工具栏、工具盒、工具架、多种面板和视图、时间滑动条、命令栏、信息栏、热盒、右键菜单等。

常用面板和对话框包括Attribute Editor（属性编辑器）、Tool Settings（工具设置）、Channel Box / Layer Editor（通道盒与层编辑器）、Outliner（概要列表）、Hypergraph（连接图超级编辑器）、UV Texture Editor（贴图坐标编辑器）、Connection Editor（连接编辑器）、Visor（资源浏览器）、Graph Editor（动画曲线编辑器）、Dope Sheet（动画表格编辑器）、Trax Editor（非线性动画编辑器）、Hypershade（着色图超级编辑器）、Render View（渲染视图）、Render Settings（渲染设置）、Dynamic Relationships（动力学关联）、Preferences（系统设置）、Expression Editor（表达式编辑器）、Script Editor（脚本程序编辑器）等。上述面板和对话框都可以在主菜单的窗口下找到。

Maya的一般操作包括二维（三维）视图操作、所选对象的变换操作（位移、旋转、缩放）、物体创建和工具设置与分步骤式交互操作、操作器、时间滑动条与动画播放、热盒与鼠标右键菜单、数值类属性拖动式修改、节点属性连接操作等。

此为试读，需要完整PDF请访问：www.ertongbook.com

二、节点

在繁多的界面、功能背后，Maya这一强大的三维动画软件是建立在一个以节点及节点之间连接为核心的系统架构上的。一个节点是一系列属性和功能的组合体，每个节点都有名字。在Maya中，不论是几何模型、编辑修改、动画、骨骼、材质、纹理、灯光、相机，还是动力学场、粒子、毛发、流体、菜单、按钮、对话框，每一个单元都是某一类型的节点。在Maya Help帮助文档的Nodes中，可以查看几百种节点的功用及其属性。可以设置关键帧动画的属性，又被称为通道。

孤立的节点是没有意义的，节点之间需要建立特定的属性连接。在Maya中，每个操作本身都在创建所需要的节点和相关连接。我们还可以在Hypergraph或者Hypershade中利用Connection Editor手动修改两个节点之间的属性连接。

在Maya中，每一个存在于三维空间中的对象是用“有向无循环图”(Directed Acyclic Graph，DAG)来表示的。形象地说，就是一种树状的层级关系。我们在Outliner中可以看到当前场景的对象列表及其层级关系。每一个对象实际上包含了两个节点：一个是定位节点（Transform Node），记录对象的位置、方向和缩放比例等三维变换信息，该对象的名称亦由此节点确定；另一个是形状节点（Shape Node），它记录了该对象用于显示或者渲染的几何信息。

对于某对象具体的形状，除了Shape Node本身可以记录之外，还可以由其他几何处理相关的节点组合连接输出到Shape Node。若由其他节点输出得到的形状，我们称此对象具有构建历史（Construction History）。

主工具栏上保持Construction History为on的状态，在Maya里建立一个多边形球体，在Hypergraph中观察该对象的输入输出连接（Input and Output Connection），可以看到它的Shape Node几何信息来源是一个polySphere类型（即多边形球体生成）的节点。我们可以在Attribute Editor或者Channel Box中找到此节点，改变该节点中Radius等参数（属性），可以看到视窗中显示的多边形球体也随之发生变化。

我们可以在Attribute Editor或者Channel Box中浏览当前选择对象所连接的节点，修改各节点名称或属性。但是，当互连的节点较多、连接网络较复杂时，使用Hypergraph（如图1—3所示）或者Hypershade来查找节点要方便得多。

如果删除该对象的History，在Hypergraph中可以看到只有该对象必有的Shape Node保留了最终结果，

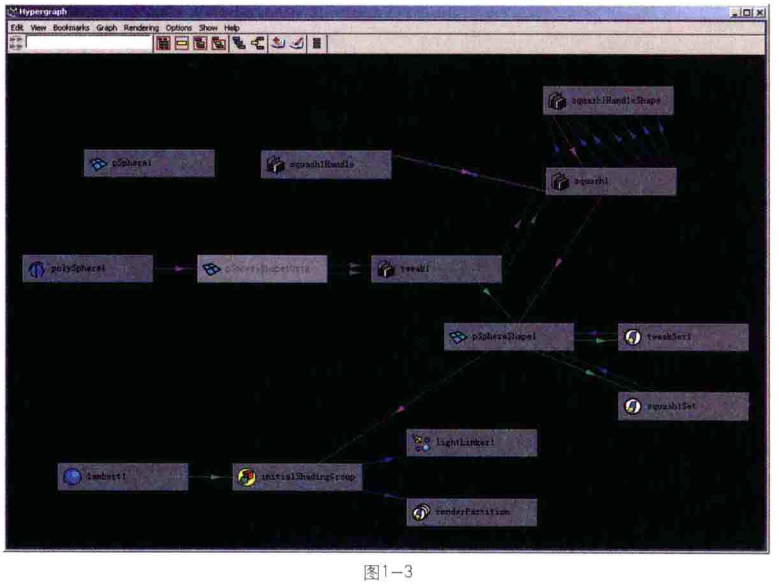

图1—3

上面便捷的修改球体大小等操作就不能进行了，这样做的好处是节约了计算机处理节点的时间和内存消耗——在复杂的三维场景中这是很突出的影响。

如果做更多的编辑变形等修改操作，该对象连接的节点数会不断增加，连接的模式也会变得复杂交织起来。这种节点连接图我们称为“依存图”（Dependency Graph，DG），显然DAG是DG的一种特例。具备History的对象，它的Shape Node的输入来源于DG中，任意节点的属性变化，或者节点间属性连接的改变，都会影响到最终的形状。

我们利用Hypershade或者Hypergraph，观察材质纹理着色的ShadingGroup可以发现，对于ShadingGroup节点，通常在它的SurfaceShader通道连接了一个材质（Material）类别的节点；可以在材质节点的Color等通道连接一个纹理（Texture）类别的节点；而一个纹理节点又可以同时输出到其他着色节点的不同通道。这些渲染着色相关的节点，可以按需要组合出复杂的Shading Net，同样属于DG。

由此，我们大略可将Maya的节点分成3大类：① 三维视图可见，可修改位置，表示某一完整对象的DAG Node。② 创建或者修改输出某对象之几何形状信息的DG Node。③ 用于组合产生渲染着色效果的DG Node。

另外，还有一些独特或者通用的DG Node，如Time（帧时同步）、AnimCurve（动画曲线）、Expression（表达式）等。

通过制作一些简单三维样例，观察每一操作步骤所添加的节点及其连接情况，查阅Help中相关的说明，就可以逐渐地理解Node及其连接图在Maya中的作用（如图1–4所示）。

三、特效

Maya中属于“渲染效果”范围的功能模块，包括Paint Effects（画笔），可以设置简单变形动画；Atmosphere（大气效果），如Environment Fog（环境雾）、Volume Shader（体积烟／雾）、Light Fog(灯光雾)；Light Effects（光效），如Optical FX镜头光效提供的Glow(光晕)、Halo(光环)

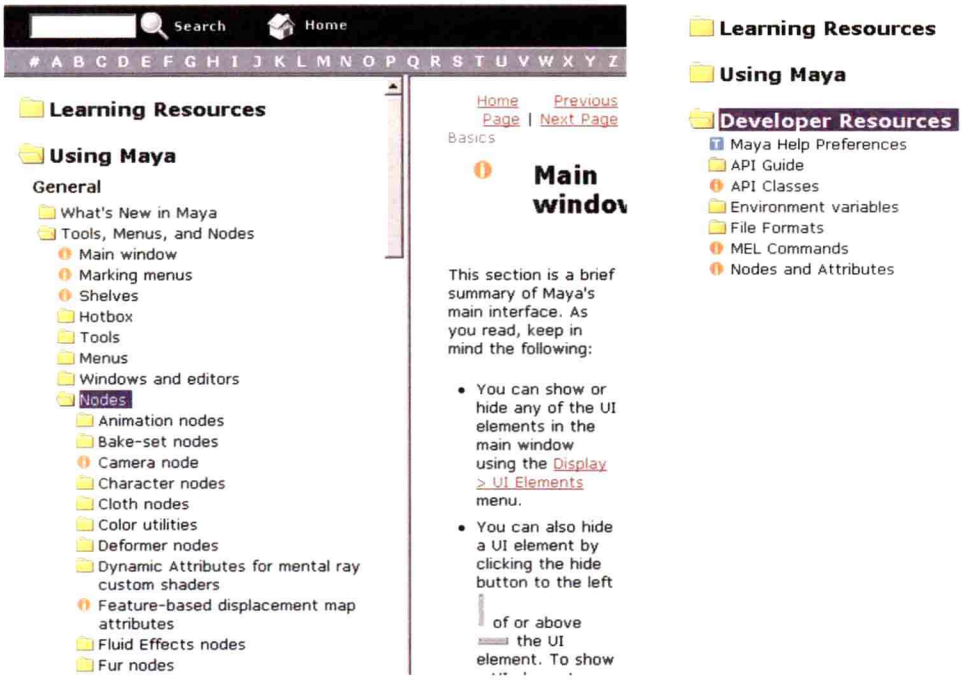

图1–4

和Lens Flare(光斑)；Toon (卡通渲染)，主要特征包括Outline/Profile (轮廓勾边) 和Tone (明暗色阶化)。

这些功能模块通常视为Maya渲染应用的一部分，本书不做详细说明。

Maya中属于"动力学特效"范围的功能模块，包括Soft/Rigid Bodies (柔体与刚体动力学)；Particles (粒子系统)，大量相同或相似对象的动态表现；Cloth/nCloth (布料动力学)；Fur，适合制作短毛发或草地；Hair，适合制作长发，可制作卷曲、扎束等发型；Fluid Effects，表现气体或者液体的流动。

通过Visor，可以浏览和应用Ocean、Fluid、Hair等范例。Maya内部预设了多个特效模板，它们在菜单栏的Effects下，用MEL程序实现参数设置界面和自动完成设置过程，大多是利用粒子系统制作效果，用表达式控制动画。如Fire（火焰）、Smoke（烟雾）、Fireworks（烟火）、Lightning（线状闪电）、Shatter（壳状或者块状碎裂）、Curve Flow（沿曲线流动的粒子）、Surface Flow（沿曲面流动的粒子）等。

本书的内容侧重于刚体动力学和粒子系统。其他部分因设置属性较多，限于篇幅和所针对的特效初学者群，仅作简要介绍和样例示范。感兴趣的读者可深入阅读Maya Help中相关部分的说明和样例（如图1—5所示）。

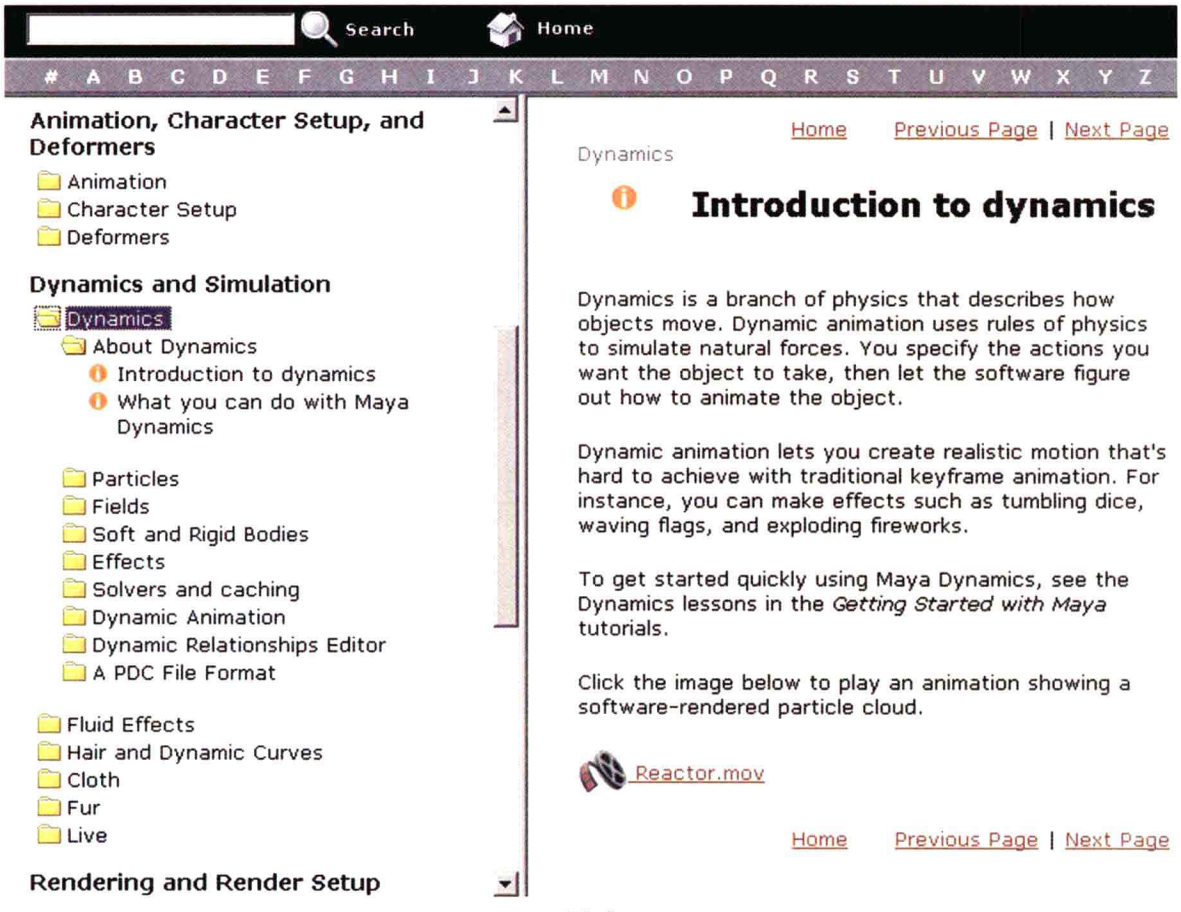

图1—5

第三节　Maya脚本语言概述

一、MEL初体验

Maya嵌入式语言(Maya Embedded Language)是专属于Maya的一种脚本解释式编程语言。如果把Maya比作一个管弦乐团，节点比作乐手，那么MEL则可以比作指挥和乐谱。自Maya软件开始启动，MEL就开始主控它的运作，一系列的MEL Script程序被载入和执行，装入各功能组件，初始化系统，加载菜单、工具架、视图……在Maya中的任何一个操作——创建、编辑，都是系统通过判别鼠标或者按键的操作以及相应的用户界面部位、调用相应MEL实际执行的。

MEL不仅能执行一系列操作命令，它本身也可以表达几何形状等场景数据。若将在Maya中制作的场景保存为文本格式（采用ASCII码记录，即.ma格式）时，用任何文本编辑器打开此ma格式文件阅读，就会发现它完全就是一个MEL Script文件。

MEL，可以用于修改缺省界面，或者自定义新的用户界面，访问高级特性；为满足建模、动画、动力学、渲染等的实际需求，将多种操作流程或者大量重复操作整合成一个简洁易用的新工具；为满足团体协作分工和项目管理的需要，开发一系列资源、对象、流程等的管理工具。

点击右下角的▣图标，打开Script Editor窗口（如图1—6所示），创建一个多边形球体，我们可以在

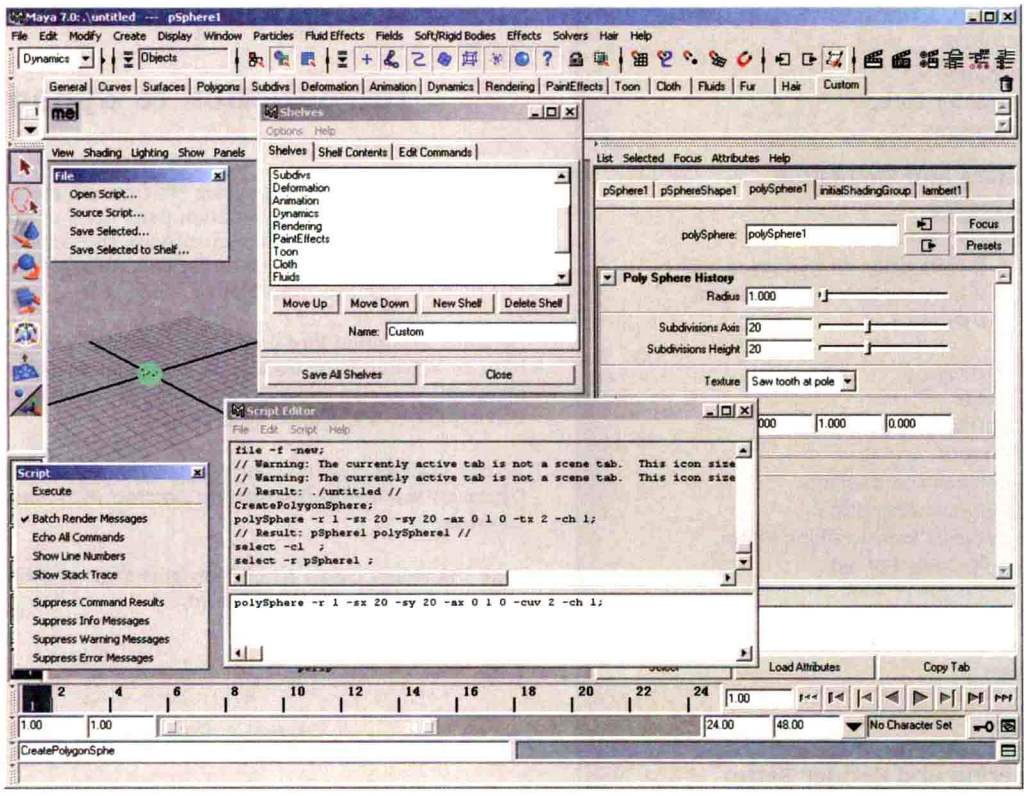

图1—6

Script Editor上端窗口中看到如下的内容：

```
CreatePolygonSphere;
polySphere -r 1 -sx 20 -sy 20 -ax 0 1 0  -tx 2
-ch 1;// Result: pSphere1 polySphere1 //
```

删除刚刚创建的球体，在Command Line或者Script Editor下端窗口输入下面代码，并按回车键以执行（在Script Editor中，须按右边数字键区的回车键）：

```
polySphere -r 1 -sx 20 -sy 20 -ax 0 1 0  -cuv
2 -ch 1;
```

可以看到Maya中又创建了一个一模一样的多边形球体，也就是说此句MEL命令就能完成创建工作。那么前一句CreatePolygonSphere是做什么的呢？其实，它是一段MEL程序定义的命令，在运行中获取当前的多边形球体缺省创建参数，然后形成后面带一系列参数的polySphere命令并执行。

我们可以在Maya Help的Commands下找到数百条MEL命令的解释，可以按字母排序或者功能分类查找。这些MEL命令是Maya中的基础命令，运行速度非常快。polySphere是其中之一。我们还可以用MEL编写定义新的命令以简化，就如同使用缺省参数的CreatePolygonSphere。

让我们再换一种方法，在Script Editor中输入下面的MEL命令：

```
createNode transform -n polyBall;
createNode mesh -n polyBallShape -p polyBall;
createNode polySphere -n polyBallCtrl;
connectAttr -f polyBallCtrl.output
polyBallShape.inMesh;
sets -e -fe initialShadingGroup polyBall;
```

上面代码运行的结果和之前的操作完全一样。每句命令逐条解释如下：

1．创建一个名为polyBall的Transform Node；

2．创建一个名为polyBallShape的多边形Shape Node，并将polyBall设为它的父级，这样就形成了一个完整的DAG对象；

3．创建一个类型为polySphere、名为polyBallCtrl的节点，用于输出半径和网格密度参数，可控制多边形球体信息；

4．将polyBallCtrl的多边形信息输出通道output连接到polyBallShape的多边形信息输入通道inMesh，此时球体已可见可调，但显示为线框；

5．将polyBall添加入initialShadingGroup（缺省材质组）这一个集合中，此步骤将polyBall设置为缺省材质。

我们可以在Script Editor中选择这些程序代码，然后用鼠标中键拖放到工具架某个标签的空位上（如图1-6所示的Custom Shelf），以后就可以点击这个新创建的图标快捷地运行代码了。点击主菜单中Settings/Preferences子菜单下的Shelves，可以在打开的窗口中观察和修改工具架上每一个分类标签Shelf之下的组员——快捷操作，以及各个快捷操作对应的文字标示、图标以及对应的MEL命令。

我们也可以将MEL代码保存为.mel的脚本程序文件，用Script Editor窗口中File下的Source Script载入。

二、MEL语法概述

MEL的语法和c/c++有点相似，学习过c/c++、Java、JavaScript的读者可以很快地熟悉它，对曾学习过其他编程语言的人也不会觉得很困难。在这里简单地介绍下MEL语言的规则和特色。

MEL是一门强类型语言，也就是说每一个变量在使用前都需要定义它的类型。变量名由一个$字符开头，后面紧跟的必须是一个英文字母或者下划线，之后则可是英文字母、数字和下划线的任意组合。例如：

```
float $x; // （浮点类型变量）
int $f32ab = 323; // （整数类型变量）
float $_init = 73.41; // （浮点变量初始化）
int $abc_321 = 0xA0; // （整数变量初始化，数值
```

```
是160的十六进制表示）
string $talk = "This is GREAT!"; //（字符串类型变量初始化）
vector $position = <<1.5, 2, 3.5>>; //（矢量类型变量初始化，由三个浮点数组成）
```

对于字符串中一些特殊的字符，需要转换码代替：

```
\"（双引号）； \n（新行）； \t（TAB间隔）
\r（回车）； \\（反斜杠）。
```

使用print命令可以显示变量、节点的属性或者运算表达式的结果。如以下的例子：

```
int $a = 5;
$a += 3; //（$a现在为8）
print $a; //（显示结果为8）
print($a + 22.3); //（显示结果为30.3）
```

注意：当要显示运算表达式结果时，须用括号将运算表达式括起。

```
/* 下面的字符串过长，可用单\字符隔断，换行续写 */
string $str = "This is a long string. \
Let's print the new \"$a\":\n";
print($str + $a + "\n"); //（字符串可用加号连接起来）
```

Maya中的数组array是可变长度的。数组元素必须是同样的类型，数字、字符串等均可，但不能是数组类型，也就是只允许一维数组。常见操作如下：

```
int $iss[]; //（创建一个空的整数数组）
$iss = {1, 3, 2}; //（设置头三个数组元素，此时数组长度为3）
$iss[5] = 15; //（索引为5时表示第六个数组元素，此时数组长度为6）
print "First of $iss[] is: ";
print $iss[0]; //（索引为0时表示第一个数组元素）
print "\nSize of $iss[] is: ";
print(size($iss)+"\n"); //（显示数组长度为6）
clear($iss); //（将数组清空）
```

矢量vector有三个浮点数分量x、y、z，常见操作有：

```
vector $a = <<0.5, 2, 1>>;
vector $b = <<2, 3, 4>>;
print("$a.x is "+ $a.x + "\n"); //（显示0.5）
print("$b.z is "+ $b.z + "\n"); //（显示4）
print("$a * 4 = " + ($a * 4) + "\n"); //（显示2、8、4）
print("$a + $b = " + ($a + $b) + "\n"); //（显示2.5、5、5）
print("$a * $b = " + ($a * $b) + "\n"); //（显示矢量点积，即0.5*2+2*3+1*4=11）
```

不可以直接给矢量变量的分量赋值，即$a.x = 23是错误的。应该写成：

```
$a = <<23, $a.y, $a.z>>;
```

不同于可变长度、一维的array，矩阵matrix是一个固定大小的二维浮点数表，常用操作有：

```
Matrix $abc[3][4] = <<1, 2, 3, 4;
                      10, 1, 1, 1;
                      0, 0, 0.1, -37>>; //（创建一个三行四列的矩阵并赋值）
$abc[1][0] = 3; //（把第二行第一列的元素值由10改为3）
```

类似c语言，MEL的语句是以;结束的。如果语句过长，可另起新行。除了变量定义、赋值、流程逻辑

此为试读，需要完整PDF请访问：www.ertongbook.com