

含

DVD
ROM

全彩印刷

内容涵盖：

渲染器概念、Maya 的内置渲染器、其他主流渲染器、VRay的渲染设置、VRay材质、VRay纹理节点、VRay灯光、VRay摄像机属性、VRay毛发、VRay环境雾、VRay卡通等。

案例包括：

雪鸟1967、桌上的玩具、快乐的周末、纹理烘焙、阳光和天空系统、渲染元素、RT引擎渲染、污垢纹理、VRay菲涅尔、IES灯光、网格灯光、圆化边、材质ID、VRay代理、毛发、VRay平面等。

陈路石 / 编著

Maya/VRay

材质与渲染技术

大揭秘

清华大学出版社



陈路石 / 编著

Maya/VRay

材质与渲染技术

大揭秘

清华大学出版社
北京

内 容 简 介

VRay 是目前业界最受欢迎的渲染引擎之一, 在国内使用比较广泛。目前这一强大的渲染引擎已经能完美支持 Maya。本书开篇首先介绍了当今各大渲染器的特点和应用, 然后对 VRay 渲染器使用范围和特点开始讲解, 接着全面介绍了 VRay 的渲染设置和纹理材质、灯光等内容, 其中使用了理论加实例结合的方式, 能让读者迅速掌握相关的知识点。本书含金量很高, 使读者能够通过一个个案例逐步掌握 VRay 渲染器相关的功能。另外, 在案例和插图的选用上注重了艺术效果。

本书适合 Maya 中高级用户学习使用, 是三维动画爱好者、影视动画制作从业人员理想的参考书, 也可作为大中院校的影视动画、广告及相关专业的教材。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Maya/VRay材质与渲染技术大揭秘 / 陈路石编著. --北京: 清华大学出版社, 2012.9

ISBN 978-7-302-28309-6

I. ①M… II. ①陈… III. ①三维动画软件, Maya、VRay IV. ①TP391.41

中国版本图书馆CIP数据核字(2012)第043875号

责任编辑: 陈绿春

封面设计: 潘国文

版式设计: 北京水木华旦数字文化发展有限责任公司

责任校对: 胡伟民

责任印制: 何 芊

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 北京世知印务有限公司

装 订 者: 三河市溧源装订厂

经 销: 全国新华书店

开 本: 203mm×260mm 印 张: 17.5 插 页: 4 字 数: 490 千字

(附 DVD1 张)

版 次: 2012 年 9 月第 1 版

印 次: 2012 年 9 月第 1 次印刷

印 数: 1~4000

定 价: 79.00 元

VRay 是由 chaosgroup 和 asgvis 公司开发的一款高质量渲染软件。VRay 是目前业界最受欢迎的渲染引擎之一，在国内使用比较广泛。目前这一强大的渲染引擎已经能完美支持 Maya 软件。VRay 有着广泛的应用领域，如计算机游戏制作、影视制作、广告设计、建筑与环境艺术表现、工业仿真、产品设计、虚拟现实等。

本书全面讲解了 VRay 渲染器的命令和相关的材质灯光等内容，在后面三章分别安排了 3 个大案例，完整讲解其渲染流程。

本书内容：

第 1 章 进入渲染的世界。作为本书的开篇，本章首先介绍了渲染的概念和算法，然后介绍了当今各大渲染器的特点和应用。

第 2 章 Maya 渲染新贵——VRay 渲染器。本章介绍了 VRay 渲染器的渲染菜单、材质、纹理节点、灯光、摄像机属性、毛发、环境雾、VRay 卡通等内容。

第 3 章 雷鸟 1967。本章通过一个汽车场景的案例主要介绍了车漆材质的应用和几种灯光的创建方法。

第 4 章 桌上的玩具。本章主要通过一个桌面玩具场景，让我们学习 VRay 的渲染的流程和各种材质的制作。

第 5 章 快乐的周末。本章通过一个室外草地画箱的场景讲解 VRay 的天光系统和使用大场景计算光子文件的方法。

本书内容安排详略得当、重点突出，理论讲解和实际案例相结合，能够快速提高读者的 Maya 的渲染技能。配套光盘提供了本书所涉及的全部场景和各类程序文件，是作者多年的经验积累，非常适合 Maya 动画的高端培训、自学和从事专业动画创作者使用。

本书由陈路石执笔编写，参与本书编写的还包括詹波、马懿、陈光福、郑丽、赵露、郑纓、王禹、刘志勇、康文、曾朝廷、陈路遥、龙利、卿洁、周宇、张彦华、陈利康、沈雪、陈丽、周利娟、陈丽、冯建华、梁纯、黄帮玉、宋子英、贺君卫、杨丽、崔洋、朱贵兰、兰小英、曾云强、蔡娜、李翠平、叶道兵、江丽、张思渠、叶友俊、林鑫、唐雪梅、刘丽娜、张侗琳、郑娟、吴娱、邹波涛、张英、陈利亚、梁奎民、陈玲薇、李淑艳、贺映梅。

由于作者水平所限，书中难免有疏漏和不妥之处，敬请广大读者批评指正。

作者

第1章 进入渲染的世界	1
1.1 渲染器概念	2
1.1.1 渲染 (Render)	2
1.1.2 渲染算法	4
1.2 Maya 的内置渲染器	7
1.2.1 Software Renderer (软件渲染器)	7
1.2.2 Mental ray 渲染器	8
1.2.3 Vector Renderer (矢量渲染器)	9
1.2.4 Maya Hardware (硬件渲染器)	10
1.3 其他主流渲染器	10
1.3.1 RenderMan (简称PRman) 渲染器	10
1.3.2 Brazil (简称BR) 渲染器	12
1.3.3 FinalRender (简称FR) 渲染器	13
1.3.4 V-Ray (简称VR) 渲染器	14
1.3.5 Maxwell渲染器	14
1.3.6 Turtle (海龟渲染器)	16
第2章 Maya渲染新贵: V-Ray渲染器	17
2.1 关于V-Ray	18
2.1.1 V-Ray简介	18
2.1.2 V-Ray的使用范围和特点	18
2.2 V-Ray 的渲染设置	20
2.2.1 加载V-Ray渲染器	20
2.2.2 V-Ray渲染器渲染设置分类	20
2.2.3 V-Ray Common (通用设置)	21
实战: 使用V-Ray Texture baking (纹理烘焙)	24
2.2.4 V-Ray选项	35
实战: 使用V-Ray Sun and sky (V-Ray阳光和天空) 系统	53
2.2.5 Indirect Illumination (间接照明)	59
2.2.6 Settings (设置)	80
2.2.7 Translator (转换)	87
2.2.8 使用Render Elements (渲染元素)	88
实战: 使用Render Elements (渲染元素)	88
2.2.9 RT Engine (RT引擎)	89
实战: 使用RT Engine (RT引擎) 进行渲染	91
2.3 V-Ray 材质	94
2.3.1 V-RayMtl (V-Ray材质)	94
2.3.2 V-RayMtl2Sided (V-Ray双面材质)	107

2.3.3	VRayLightMtl (VRay灯光材质)	109
2.3.4	VRay Mtl Wrapper (VRay包裹材质)	111
2.3.5	VRayFastSSS2 (VRay快速次表面散射材质)	116
2.3.6	VRayBlendMtl (VRay混合材质)	121
2.3.7	VRaySimbionMtl	123
2.3.8	VRayMeshMaterial (VRay网格材质)	125
2.3.9	VRayBumpMtl (VRay凹凸材质)	126
2.3.10	VRayCarPaintMtl (VRay车漆材质)	128
2.4	VRay 纹理节点	134
2.4.1	VRayDirt (VRay污垢纹理)	134
	实战: VRayDirt (VRay污垢纹理)	137
2.4.2	VRayEdges (VRay边)	138
2.4.3	VRay Water (VRay海洋)	139
2.4.4	VRayFresnel (VRay菲涅尔)	140
	实战: VRayFresnel (VRay菲涅尔)	140
2.5	VRay 的灯光	141
2.5.1	VRayLightSphere (VRay球形灯)	141
2.5.2	VRay Light Dome (VRay半球灯)	144
2.5.3	VRayLightRect (VRay矩形灯光)	145
2.5.4	VRayLightIES (VRay ies灯光)	145
	实战: VRayLightIES (VRay ies灯光)	146
2.5.5	VRayLightMesh (VRay网格灯光)	146
	实战: VRayLightMesh (VRay网格灯光)	146
2.6	VRay 扩展属性	148
2.6.1	摄像机属性	148
2.6.2	变换属性	156
2.6.3	网格属性	157
	实战: Round Edges (圆化边)	159
2.6.4	灯光属性	159
2.6.5	材质属性	160
2.6.6	Shading Engine Extra Attributes (光影组扩展属性)	160
	实战: MaterialID (材质ID)	160
2.7	杂项节点	162
2.7.1	VRayProxy (VRay代理)	162
	实战: VRayProxy (VRay代理)	163
2.7.2	VRayFur (毛发)	168
	实战: VRayFur (毛发)	172
2.7.3	VRayEnvironmentFog (VRay环境雾)	176
2.7.4	VRayPlane (VRay平面)	178

实战: V-RayPlane (V-Ray平面)	179
2.7.5 V-RayToon (V-Ray卡通)	180
第3章 雷鸟1967	183
3.1 导入场景文件	184
3.1.1 渲染最终效果	184
3.1.2 打开场景文件	185
3.2 创建场景灯光和环境照明	185
3.2.1 创建主灯光	185
3.2.2 创建辅助灯光	186
3.2.3 创建车灯照明	186
3.3 设置场景材质	187
3.3.1 制作外景材质	187
3.3.2 制作汽车材质	191
3.4 渲染设置	205
3.4.1 最终效果	207
第4章 桌上的玩具	208
4.1 导入场景文件	209
4.1.1 渲染最终效果	209
4.1.2 打开场景文件	209
4.2 创建场景灯光和环境照明	209
4.2.1 创建主灯光	209
4.2.2 创建辅助灯光1	210
4.2.3 创建辅助灯光2	210
4.2.4 创建辅助灯光3	211
4.3 设置场景材质	212
4.3.1 制作房屋材质	212
4.3.2 制作房屋周围材质	215
4.3.3 制作木桶材质	220
4.3.4 制作油灯材质	221
4.3.5 制作轨道材质	223
4.3.6 制作锅炉材质	224
4.3.7 制作雕像材质	232
4.3.8 制作木板材质	233
4.3.9 制作梯子材质	233
4.3.10 制作机器材质	234
4.3.11 制作地面材质	240

4.3.12	制作木棍材质	240
4.4	渲染设置	243
第5章 快乐的周末		245
5.1	导入场景文件	246
5.1.1	渲染最终效果	246
5.1.2	打开场景文件	246
5.2	创建场景灯光	246
5.2.1	创建阳光和天光	246
5.2.2	启用GI全局光照	247
5.3	设置场景材质	248
5.3.1	制作草地材质	248
5.3.2	制作木板1材质	249
5.3.3	制作石头材质	250
5.3.4	制作线材材质	251
5.3.5	制作画箱材质1	252
5.3.6	制作画箱材质2	253
5.3.7	画箱金属材质	254
5.3.8	画箱金属材质2	254
5.3.9	制作合页材质	255
5.3.10	制作显示器材质	256
5.3.11	制作画板材质	257
5.3.12	制作画板材质2	258
5.3.13	制作架子材质	259
5.3.14	制作颜料盒材质	259
5.3.15	制作颜料材质	260
5.3.16	制作画笔材质1	264
5.3.17	制作画笔材质2	265
5.3.18	制作画笔材质3	265
5.3.19	制作画笔材质4	266
5.3.20	制作图钉、螺丝、画笔金属、回形针、画刀金属材质	266
5.3.21	制作铅笔材质	267
5.4	渲染设置	268
5.4.1	计算光子文件	268
5.4.2	最终渲染设置	270
5.4.3	最终效果	270

第1章 进入渲染的世界

本章学习要点：

- 掌握渲染概念
- 了解渲染器的几种算法
- 认识 Maya 内置的几种渲染器
- 介绍当今几种主流渲染器的特点和应用

渲染是一个充满着趣味的过程，它能让我们心中所设想的五彩缤纷的 CG 世界直接展示在您的面前，在现实世界中无法实现的甚至不可能的景观都能在这里实现。

无论是静帧还是动画，渲染都是其中最重要的阶段。因为无论制作什么作品，最终都是要将它拿给别人看的，如果我们有好的情节、好的模型、好的动画，而渲染的效果却很糟糕，那确实是很遗憾的；相反，如果拥有好的渲染表现，就会相当有成就感。

本章将先讨论一些渲染方面的基础知识，让大家对渲染有一个感性认识。本章会讲到几种常见的渲染算法，它们其实就是渲染器渲染出图的基本原理，这里只需要了解即可，不需要去研究。理解一些理论知识对于以后相关的渲染工作会有很大帮助。接下来将一起来看看 Maya 内置的几种渲染器的设置及工作原理，为后续内容的展开做一个铺垫。最后还要介绍当今主流的几种渲染器的特点和应用范围，便于有针对选择渲染器进行应用。

1.1 渲染器概念

1.1.1 渲染 (Render)

Render 就是经常所说的“渲染”，直译为“着色”，也就是为场景对象进行着色的过程。当然这并不是简单的着色过程，计算机会经过相当复杂的运算，将虚拟的三维场景投影到二维平面上，从而形成最终输出的画面，如图 1-1 所示。



图1-1

也有人把它称为 Shade (着色)。Render 和 Shade 两个词在三维软件中是截然不同的两个概念，虽然它们的功能很相似，但却有不同。Shade 是一种显示方案，一般出现在三维软件的主要窗口中，和三维模型的线框图一样起到

辅助观察模型的作用。

很明显，着色模式比线框模式更容易理解模型的结构，但它只是简单的显示而已，数字图像中把它称为“明暗着色法”，如图 1-2 所示。

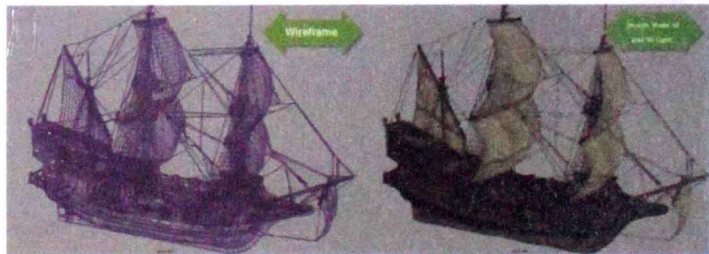


图1-2

在像 Maya 这样的高级三维软件中，还可以用 Shade 显示出简单的灯光效果、阴影效果和表面纹理效果，如图 1-3 所示。



图1-3

当然，高质量的着色效果是需要专业三维图形显示卡来支持的，它可以加速和优化三维图形的显示。所以，在作图时显卡的选择也是

第1章 进入渲染的世界

非常重要的。现在显卡换代速度非常快，往往刚买的显卡过不了几个月就会落后。不过建议大家购买显卡以实用为主，因为普通的显卡又称“游戏显卡”，与专业做图的专业显卡差别是很大的，所以经常有人说，为什么花一、两千元购买的显卡（例如 8800gts、HD3850 等），在使用某些专业做图软件（3dsMax、Maya 等）时经常出现问题，所以价格很贵的游戏显卡甚至还不如落后它几代的专业显卡。所以普通游戏显卡，主要针对游戏、家庭娱乐领域，并不适合专业做图领域。而专业显卡如果应用在游戏中，其速度也比不上相同规格的游戏显卡。作为一般的使用者而言，普通游戏卡也能满足我们的要求。

专业显卡的优势在于模型面数很多的情况下，视图仍然能够流畅的移动，并且能够真实、正确的再现赋予了纹理的模型。此外，很多设计软件还针对专业显卡做了优化，能实现一些特定的功能。通过如图 1-4 所示可以看到，专业显卡的外观和游戏卡也是有区别的。左图是专业显卡，右图是游戏显卡。

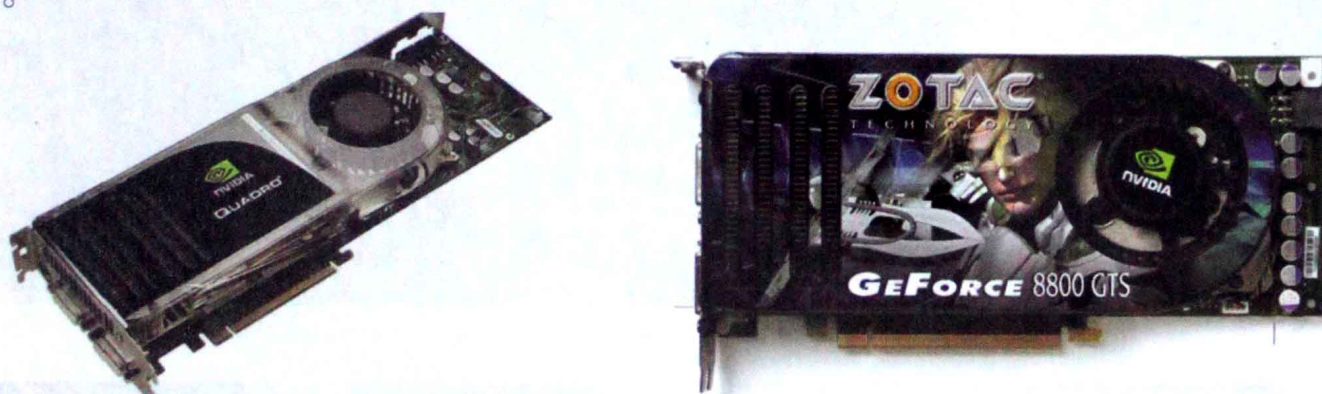


图1-4

下面是同等级别的专业显卡和游戏显卡的专业软件测试得分，可以看到专业显卡远远超过了游戏显卡的得分，如图 1-5 所示。

```
----- SUM_RESULTS\3DSMAX\SUMMARY.TXT
3dsmax-03 Weighted Geometric Mean = 61.42

----- SUM_RESULTS\CATIA\SUMMARY.TXT
catia-01 Weighted Geometric Mean = 50.16

----- SUM_RESULTS\ENSIGHT\SUMMARY.TXT
ensight-01 Weighted Geometric Mean = 33.00

----- SUM_RESULTS\LIGHT\SUMMARY.TXT
light-07 Weighted Geometric Mean = 40.00

----- SUM_RESULTS\MAYA\SUMMARY.TXT
maya-01 Weighted Geometric Mean = 105.7

----- SUM_RESULTS\PROE\SUMMARY.TXT
proe-03 Weighted Geometric Mean = 67.22

----- SUM_RESULTS\SW\SUMMARY.TXT
sw-01 Weighted Geometric Mean = 40.71

----- SUM_RESULTS\UGS\SUMMARY.TXT
ugs-04 Weighted Geometric Mean = 35.45
```

```
----- SUM_RESULTS\3DSMAX\SUMMARY.TXT
3dsmax-03 Weighted Geometric Mean = 45.77

----- SUM_RESULTS\CATIA\SUMMARY.TXT
catia-01 Weighted Geometric Mean = 29.94

----- SUM_RESULTS\ENSIGHT\SUMMARY.TXT
ensight-01 Weighted Geometric Mean = 28.72

----- SUM_RESULTS\LIGHT\SUMMARY.TXT
light-07 Weighted Geometric Mean = 32.61

----- SUM_RESULTS\MAYA\SUMMARY.TXT
maya-01 Weighted Geometric Mean = 63.80

----- SUM_RESULTS\PROE\SUMMARY.TXT
proe-03 Weighted Geometric Mean = 53.70

----- SUM_RESULTS\SW\SUMMARY.TXT
sw-01 Weighted Geometric Mean = 31.41

----- SUM_RESULTS\UGS\SUMMARY.TXT
ugs-04 Weighted Geometric Mean = 35.06
```

图1-5

但无论怎样优化,它和CPU处理器渲染出来的效果还是有不少差距的,但是差距已经在慢慢缩小,这是因为 Shade 采用的是一种实时显示技术。在几年前实时渲染还无法反馈出场景中的反射、折射等光线追踪效果。随着硬件的飞速发展以及更高级的管理和渲染应用的发展,更高的复杂度变得可能,以往要通过 CPU 渲染才能完成的工作,通过显卡实时渲染的效果已经慢慢接近于 CPU 渲染的效果。

如图 1-6 ~ 图 1-8 所示为历年的游戏显卡实时 Dmimo 渲染展示截图,分别代表了当时实时渲染技术的最高水平,可以观察到后面几张截图效果已经比较真实了。



图1-6



图1-7

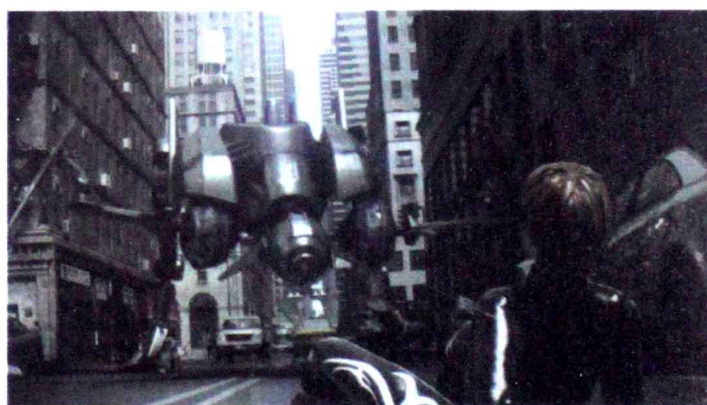


图1-8

技巧与提示:

渲染可以分为实时渲染和非实时渲染。实时渲染可以实时地将三维空间中的内容反应到画面上,能即时的计算出画面内容,如上面提到的游戏画面就是实时渲染;非实时渲染是将三维作品提前输出为二维的画面,再将这些二维画面按一定速率进行播放,如电影、电视等都是非实时渲染。

1.1.2 渲染算法

从渲染的原理来看,可以将渲染的算法分为 Scanline (扫描线算法)、Raytrace (光线追踪算法) 和 Radiosity (热辐射算法),每种算法都有其存在的意义,因为它们并不是独立存在的。

第1章 进入渲染的世界

1.Scanline (扫描线)

扫描线 (Scanline) 是最早被开发出来的渲染方法, 它发展至今已经相当成熟, 也是现在运用最多的渲染方式。扫描线优点是速度很快, 缺点是不能计算反射和折射效果, 其中 Pixar 的 RenderMan 渲染器最具代表性, RenderMan 的 REYES (Render Everything You'd Ever Seen) 算法是 Scanline (扫描线算法) 的最极致发挥, 如图 1-9 所示。

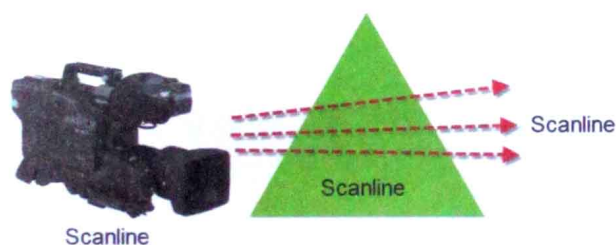


图1-9

使用 Scanline (扫描线) 方式渲染的图片, 如图 1-10 所示。

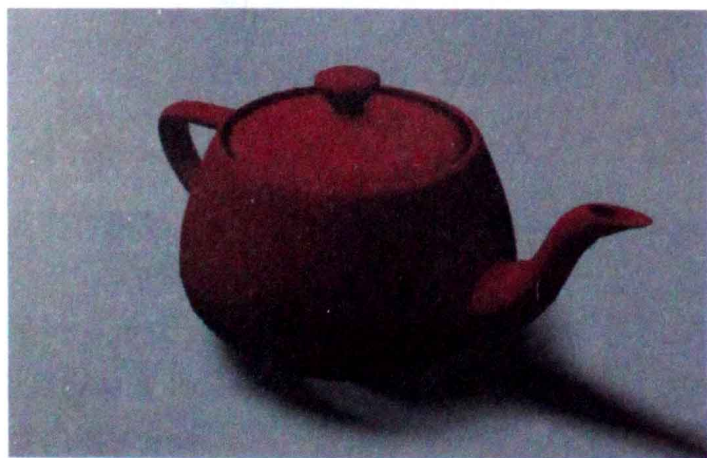


图1-10

2.Raytrace (光线追踪)

光线追踪 (Raytrace) 是最早开发出来, 解决反射、折射及非漫射表面光能传递的, 是生成高质量画面的渲染算法之一, 能实现逼真的反射和折射效果, 如金属玻璃类物体。

Raytrace (光线追踪) 的定义: 光是从

摄影机投射到场景中, 当光与物体发生碰撞时会再反弹或折射, 并产生出更多其他的光线。由于 Raytrace (光线追踪) 是属于递归算法, 其核心为每次交集后用自身再次产生交集, 因此在渲染时需要相当长的渲染时间, 为了解决 Raytrace (光线追踪) 在速度上的问题, Mentalray BSP (Binary Space Partition) 二进制空间交叉算法来提升渲染的速度, 如图 1-11 所示。

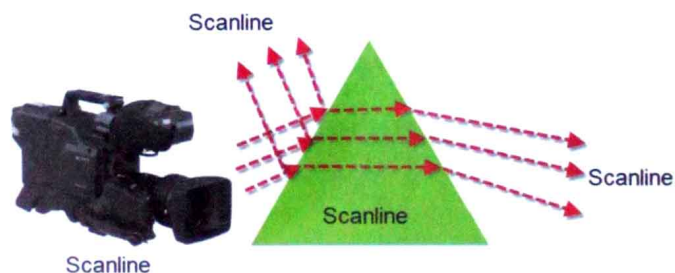


图1-11

光线追踪算法会进行庞大的信息处理, 与扫描线算法相比, 其速度相对较慢, 但可以产生真实的反射和折射效果。由于 Raytrace (光线追踪) 假设光线是一条没有大小的 Ray (射线), 通常造成 Raytrace (光线追踪) 无法很精确地采样到漫反射表面上的灯光信息, 所以就产生了另一种渲染方法 Radios (热辐射) 来做弥补。

3.Radiosity (热辐射)

反射与折射的传递方式, 但对日常生活中最常见的漫反射效果则显得束手无策。后来学者把热力学的热辐射方法引入了光能的传递运算, 成功模拟了漫射面之间的光能传递, 这就是——Radiosity (热辐射) 算法, 例如, 物体表面颜色所产生的相互渗透或是溢色效果。热辐射算法是基于热辐射能在物体表面之间的能量传递和能量守恒定律。热辐射算法可以使光线在物体之间产生漫反射效果, 直至能量耗尽。热力辐射的方式是扩散式的, 而光能在漫射表

面上的反射也可以视为一种扩散方式。

这种算法可以使物体之间产生色彩溢出现象,能实现真实的漫反射效果。如图 1-12 所示。

因此热辐射在计算时并不会受到每条光线的限制,它反而可以把能量一一分送出去,并让能量在一个封闭的环境对物体表面产生多重反射,直到能量达到平衡为止。Radiosity (热辐射) 渲染基于物理学理论,其渲染效果真实,是 Raytrace (光线追踪) 所不能比拟的,但从视觉效果上考虑,现在 Raytrace (光线追踪)

和 Radiosity (热辐射) 不相上下,在速度上, Raytrace (光线追踪) 更占绝对优势。而且, Refract (折射)、Caustic (焦散效果) 是 Radiosity (热辐射) 无法模拟的。

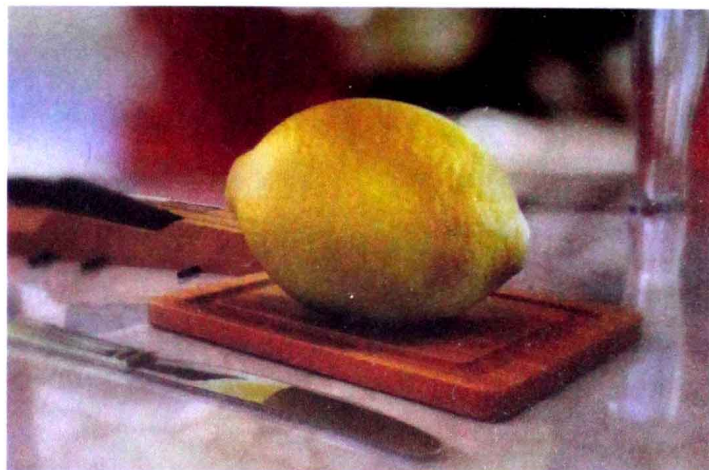


图1-12

技巧与提示:

如图1-13所示中左图只使用Raytrace (光线追踪) 方式,右图使用Raytrace (光线追踪)、Radiosity (热辐射) 两种算法相结合的方式。右图让光能在物体表面之间来回传递,得到了比较好的效果。



图1-13

实际上这三种算法在渲染时,并非是独立来计算的,它们可以相互融合使用,并可以利用各自算法的优点来弥补其他算法的不足。在后面所介绍的渲染器中,这三种算法都是相互融合的。

以上的算法解决了 3 种光线传递方式,另外还有一种现象,这就是著名的 Caustic (焦散) 现象。因为光在经过非漫反射面的传递后会出现集中聚焦、二次光源等现象,而一般的 Raytrace (光线追踪) 和 Radiosity (热辐射) 都不能解决,于是诞生了双向光线跟踪。因为光线应该是从光源出发射向物体的,但 Raytrace (光线追踪) 光线则从屏幕处发射向物体再回溯到光源,所以一般的 Raytrace (光线追踪) 被称为“逆向光线跟踪”。这样的好处是避免不必要的光线求解,简化算法,减少运算量。但这样根本不利于光能传递的求解。双向光线跟踪的思想是在逆向光线跟踪的同时进行正向的光线跟踪。实际上双向光线跟踪是通过一张 PhotonMap (光子贴图) 来实现的。

PhotonMap (光子贴图) 是一张抽象的贴图,它包裹在物体的表面上,当进行正向光线跟踪时从光源到物体表面进行有限的光线采样,光线交在物体上,但采样结果表现在 PhotonMap (光子贴图) 上,也就是说光线是射在 PhotonMap (光子贴图) 上的。PhotonMap (光子贴图) 上的信息是光线

第1章 进入渲染的世界

信息，所以通过整理后可以得到精细的 Caustic（焦散）贴图，在渲染时要得到物体表面信息可以直接从 PhotonMap（光子贴图）上得到相应的面上的贴图信息。同理，使用 PhotonMap（光子贴图）的全局光照也是通过这样的采样方法达到光能传递的效果。PhotonMap 用在 Caustic 上是一种十分精确的方法，但在全局光照中就属于基于视觉上的算法，并不精确真实。

如图 1-14 所示为 Mental Ray 渲染器模拟焦散的效果。



图1-14

1.2 Maya 的内置渲染器

1.2.1 Software Renderer（软件渲染器）

Maya 内置的 Software Renderer（软件渲染器）是 Maya 使用的默认渲染方式，它能提供高质量渲染图像，将光线跟踪和扫描线算法进行结合并支持多线程的高级渲染器，它可以流畅地完成很多工作。它的优点是不受硬件的限制，运算速度比较快。同时，它还支持 Render Farm（渲染农场）的网络渲染模式，这将会大大地提升整体的渲染效率，有效解决渲染的速度问题。它的兼容较好，当有些 Maya 的特征不被其他渲染器支持或支持得不好时，它就是我们最佳的选择。如图 1-15 所示是软件渲染器界面。

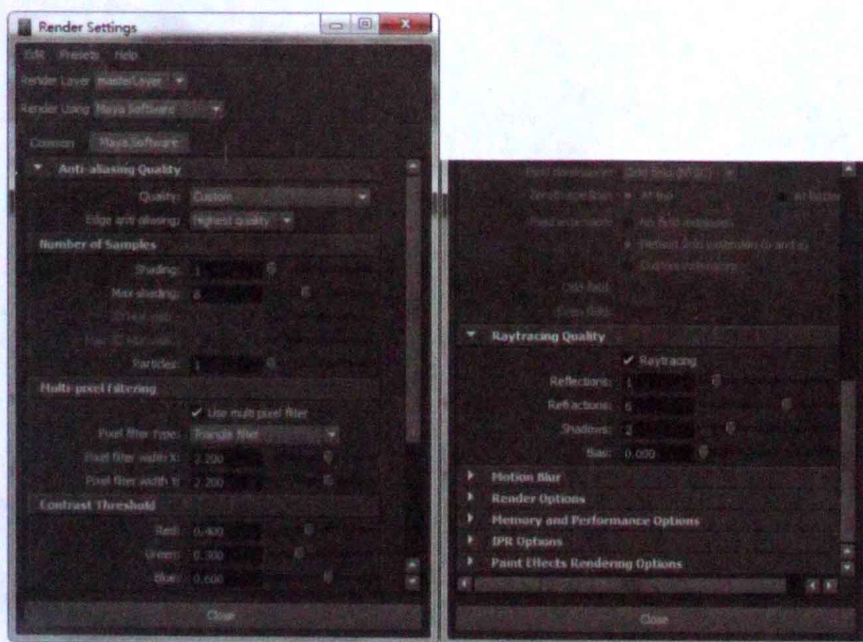


图1-15

如图 1-16 图所示为两张使用 Software Renderer (软件渲染器) 渲染的图片, 通过合理的灯光和渲染设置也能得到很好的效果。



图1-16

1.2.2 Mental ray 渲染器

Mental ray 渲染器在计算机的三维领域可以说是无人不知, 无人不晓, 它与 Pixar Renderman 并驾齐驱, 被公认为世界顶级的电影级渲染工具, 其庞大的用户群和广泛的技术支持, 是 FinalRender、Brail 等渲染新贵所远远不及的, 它能支持一些更为高级的运算功能, 提供更为逼真的图像质感。

大名鼎鼎的德国渲染器——Mental ray, Mental ray (“Mental” 的中文意思为: 精神的、脑力的; “ray” 的中文意思为: 光线、射线) 是德国的 Mental image 公司 (注 Mental image 现已成为 NVIDIA 公司的全资子公司) 最引以为荣的产品, 被称为渲染领域的 “德国战车”。Mental ray 是一款超强的高端渲染器。它是除了 Pixar Renderman 渲染器之外拥有最广泛用户的电影级渲染工具。其庞大的用户群体和广泛的技术支持, 是远非 Final Render 和 Barzil 这一类渲染新贵可比, 如图 1-17 所示是该渲染器的操作界面。

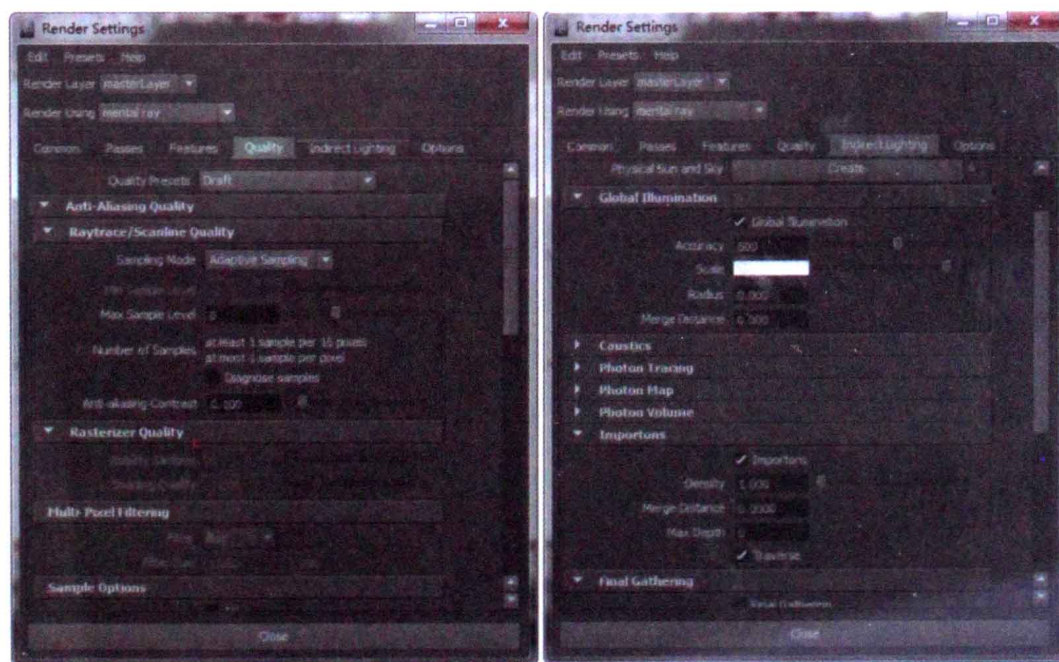


图1-17