



游戏开发经典丛书

Mathematics for
3D Game Programming & Computer Graphics

3D 游戏与计算机图形学中的 数学方法

[美] Eric Lengyel 著
詹海生 李广鑫 王鸿利 译

— 3D 游戏引擎程序设计和计算机图形学的核心数学算法

— 现代实时绘制技术

— OpenGL 架构环境下的典型应用

— 游戏中物理建模的若干重要主题



清华大学出版社

前 言

本书介绍了开发专业级三维图形引擎所需的数学方法。为了对一些技术主题进行完整阐述,以及对游戏设计者使用数学工具内部的计算方法进行深入了解,本书对一些重要结论的推导进行了详细说明。

本书中所介绍的大部分内容与渲染图像所用的三维图形系统无关,但读者应熟悉使用三维图形库的基本原理,并且掌握如何由顶点和多边形构造模型的方法。在书中的一些地方,为了说明如何用图形库产生某些特殊效果,用到一些 OpenGL 中的实现方法。在第 6 章中,提供了执行特殊光照计算的不同顶点程序。这些程序使用了现代图形硬件支持的向量指令集,并且遵循 OpenGL 的扩展函数 GL_NV_vertex_program 规范。

本书每一章的结尾都对该章中出现的重要方程和公式进行了总结,用以作为快速参考工具。读者利用它们可以快速找到结果,而不必将该结果的推导过程一一阅读完。每一章还提供了一些习题,其中需要计算的习题答案在附录 E 中给出。

内容概要

第 1 章:向量 首先回顾了向量的定义及其性质。向量是研究三维计算机图形学的基本要素。书中用到了大量的向量运算,如点积和叉积。

第 2 章:矩阵 对矩阵的掌握是三维游戏编程中另一基本要素。本章讨论了一些矩阵的基本理论,如线性方程组的矩阵表示,以及一些高级知识,如特征向量和矩阵的对角化问题。

第 3 章:变换 本章以矩阵为工具进行变换操作,如平移、旋转以及比例缩放。同时介绍了四维齐次坐标理论,该理论广泛地用于三维图形系统中不同坐标系之间的变换。此外还介绍了四元组的性质以及作为变换根工具的用途。

第 4 章:三维引擎中的几何 从本章开始,我们才看到前 3 章中介绍的方法应用到三维游戏编程和计算机图形学的实际问题的求解。在分析了三维空间中的面和线后,本章介绍了视平载体以及它与虚拟相机(即观察点)之间的关系,探讨了包含视野、投影校正插值以及投影矩阵在内的一些内容。

第 5 章:光线跟踪 光线跟踪算法在游戏编程的许多领域得到应用,如 Light-map 光照图的生成、视线判断以及碰撞检测等。本章首先介绍了计算方程的解析根和数值根的技术,然后给出了光线与一般几何体的求交方法,最后讨论了光反射与折射的计算方法。

第 6 章:光照 本章讨论了与照明和阴影相关的众多主要内容。首先列举了各种各样的光源,然后介绍了简单反射模型,接着分析了描述渲染表面细节的技术,即纹理映射、光泽映射以及凹凸映射。最后,详细解释了 Cook-Torrance 物理光照模型。

第 7 章:可视性判断 三维引擎的性能主要决定于其判断场景中哪些内容是可见的能力。本章介绍了构造不同的边界体的方法以及与视平载体之间的可视性判断方法、应用于大规模可视性判断的空间分割法和门系统法等。

第 8 章:碰撞检测 在游戏世界中,碰撞检测对于不同对象之间的交互是必需的。本章提供了检测运动物体与静止环境之间以及运动物体之间的碰撞检测方法。

第 9 章:多边形技术 本章提供了操纵多边形模型的几个常用技术,如在任意表面进行贴图的技术、相应的顶点深度补偿方法。还介绍了产生不同特效的公告牌技术、利用模板缓冲区的阴影产生技术以及多边形化简技术。

第 10 章:线性物理运动 从这一章开始本书用两章的篇幅介绍经典物理学中与运动相关的内容。三维游戏中的对象都是通过运动来展示自己的。本章首先介绍了位置函数以及二阶微分方程的解,然后分析了真空中和有阻力空间中的抛物运动,最后介绍摩擦力对运动的影响。

第 11 章:旋转物理运动 本章继续阐述运动中更高级的问题——旋转运动。首先介绍了旋转系统中物体受到的力,然后分析刚体运动并利用惯性张力推导出角速度与角动量之间的关系,最后简单介绍了弹簧和钟摆的振荡运动。

第 12 章:流体仿真 本书最后给出了基于二维波动方程的流体运动的物理模型,并提供了计算表示流体表面的规则栅格中一个顶点位置的方法,同时讨论了解收敛的必要条件。

附录 A:复数 介绍了复数的几个基本原理以及本书用到复数的几个性质。

附录 B:三角恒等式 给出了三角函数以及本书中用到的几个三角等式和三角恒等式。

附录 C:坐标系 简单回顾了圆柱坐标系和球坐标系。这些坐标系的应用出现在本书的几个地方,在第 11 章用的最多。

附录 D:泰勒级数 给出了泰勒级数的推导过程以及多个常用函数的幂级数。

附录 E:习题答案 给出了每章节末习题中的计算题的答案。

符号与约定

斜体罗马字母和希腊字母表示标量。粗斜体字母表示向量、四元组和矩阵。下表列出了本书中用到的其他符号与约定。

量/操作	符号/举例
标量	斜体字母,如 x, t, A, α, ω
角度	斜体希腊字母,如 θ, ϕ, α
向量	粗斜体字母,如 $\mathbf{V}, \mathbf{P}, \mathbf{x}, \boldsymbol{\omega}$
四元组	粗斜体字母,如 $\mathbf{q}, \mathbf{q}_1, \mathbf{q}_2$
矩阵	粗斜体字母,如 \mathbf{M}, \mathbf{P}
向量的模	双竖杠,如 $\ \mathbf{P}\ $
四元组的共轭	上杠,如 $\bar{\mathbf{q}}$
矩阵转置	上标 T,如 \mathbf{M}^T
矩阵行列式	$\det \mathbf{M}$ 或者单竖杠,如 $ \mathbf{M} $
时间的导数	点,如 $\frac{d}{dt} \mathbf{x}(t) = \dot{\mathbf{x}}$
实数集合	\mathbf{R}

续表

量/操作	符号/举例
复数集合	\mathbb{C}
四元组集合	\mathbb{H}

关于本书作者

Eric Lengyel 是美国加利福尼亚州 Redwood 城 3DO 公司资深软件工程师,他从 Virginia 技术学院获得数学硕士学位,已经为包括 gamasutra.com 在内的多个工业期刊撰写了许多论文。他还负责编写了“Game Programming Gems 2”一书中的几何管理部分。

关于本书特色

- ◆ 集中介绍了三维游戏引擎设计与计算机图形学中的重要数学方法。
- ◆ 由于 OpenGL 的跨平台特性以及长期以来被业界广为接受,本书给出了在 OpenGL 结构中的应用例程,同时还参考一些现代三维图形硬件,如 Nvidia 的 Geforce 3。
- ◆ 介绍了四元组、齐次坐标、光线跟踪、凹凸映射、门系统、多边形技术以及阴影技术和物理运动等主题。
- ◆ 讨论了现代流行的实时渲染技术。
- ◆ 给出了在 OpenGL 结构中的应用例程。
- ◆ 介绍了物理学中与三维游戏引擎设计相关的方法。
- ◆ 包含习题集,可作为教材使用。

目 录

第 1 章 向量	1
1.1 向量的性质	2
1.2 点积	4
1.3 叉积	7
1.4 向量空间	10
1.5 本章小结	12
习题	13
*	
第 2 章 矩阵	15
2.1 矩阵的性质	16
2.2 线性方程组	18
2.3 逆矩阵	21
2.4 行列式	25
2.5 特征值与特征向量	28
2.6 对角化	30
2.7 本章小结	33
习题	34
第 3 章 变换	37
3.1 通用变换	38
3.1.1 正交矩阵	39
3.1.2 手向性	39
3.2 缩放变换	40
3.3 旋转变换	41
3.3.1 绕任意轴旋转	42
3.4 齐次坐标	43
3.4.1 四维变换	43
3.4.2 点向量和方向向量	44
3.4.3 坐标的几何解释	45
3.5 法向量的变换	45
3.6 四元数	46
3.6.1 四元数数学	46
3.6.2 四元数的旋转	47
3.6.3 球型线性插值	50

3.7 本章小结	52
习题	54
第 4 章 三维引擎中的几何	55
4.1 3D 空间中的直线	56
4.1.1 点到直线的距离	56
4.1.2 两条直线之间的距离	56
4.2 3D 空间中的平面	58
4.2.1 直线与平面相交	59
4.2.2 三个平面相交	60
4.2.3 平面的变换	62
4.3 视截体	63
4.3.1 视域	63
4.3.2 视截面	64
4.4 透视校正插值	66
4.4.1 深度插值(depth interpolation)	66
4.4.2 顶点属性的插值	68
4.5 投影	68
4.5.1 透视投影	69
4.5.2 正投影	71
4.5.3 提取视截面	72
4.6 本章小结	73
习题	75
第 5 章 光线跟踪	77
5.1 根的求解	78
5.1.1 二次多项式	78
5.1.2 三次多项式	78
5.1.3 四次多项式	80
5.1.4 牛顿-拉弗森迭代法	81
5.1.5 倒数与平方根的精化	82
5.2 曲面交点	83
5.2.1 光线与三角形相交	83
5.2.2 光线与长方体相交	85
5.2.3 光线与球体相交	85
5.2.4 光线与圆柱面相交	87
5.2.5 光线与环形圆纹曲面相交	87
5.3 法向量的计算	88
5.4 反射向量和折射向量	89

5.4.1 反射向量的计算	89
5.4.2 折射向量的计算	90
5.5 本章小结	92
习题	93
第6章 光照	95
6.1 RGB 颜色系统	96
6.2 光源	96
6.2.1 环境光	97
6.2.2 定向光源	97
6.2.3 点光源	97
6.2.4 聚焦光源	97
6.3 漫反射光	98
6.4 纹理映射	99
6.4.1 标准纹理图	99
6.4.2 投影纹理图	100
6.4.3 立方体纹理图	102
6.5 镜面反射光	103
6.6 发射光	105
6.7 明暗处理	106
6.7.1 法向量的计算	106
6.7.2 Gouraud 明暗处理	107
6.7.3 Phong 明暗处理	107
6.8 凹凸映射	108
6.8.1 凹凸映射图的构造	108
6.8.2 切线空间	109
6.8.3 切向量的计算	109
6.8.4 实现	111
6.9 物理反射模型	113
6.9.1 双向反射分布函数	113
6.9.2 Cook-Torrance 光照模型	117
6.9.3 Fresnel 系数	117
6.9.4 微平面分布函数	119
6.9.5 几何衰减系数	121
6.9.6 实现	122
6.10 本章小结	126
习题	129

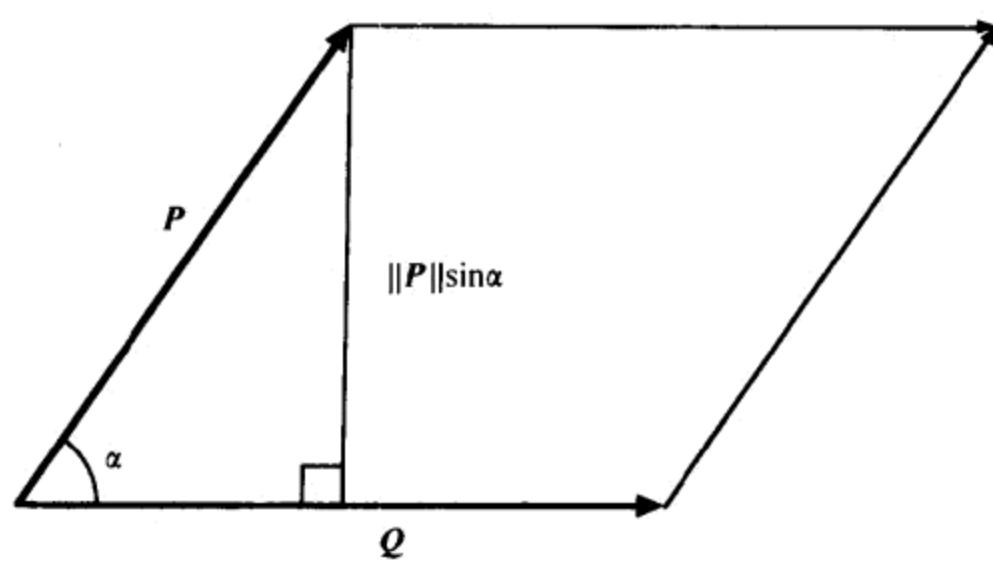
第 7 章 可视性判断	131
7.1 边界体构造	132
7.1.1 主分量分析	132
7.1.2 构造边界盒	134
7.1.3 构造边界球	135
7.1.4 构造边界椭球	136
7.1.5 构造边界圆柱	137
7.2 边界体测试	138
7.2.1 边界球测试	138
7.2.2 边界椭球测试	139
7.2.3 边界圆柱测试	141
7.2.4 边界盒测试	143
7.3 空间分割	144
7.3.1 八叉树	145
7.3.2 二叉空间分割树	146
7.4 门系统	147
7.4.1 门裁剪	148
7.4.2 缩小视平截体	150
7.5 本章小结	151
习题.....	154
第 8 章 碰撞检测	155
8.1 环境碰撞	156
8.1.1 球体与平面的碰撞	156
8.1.2 长方体与平面的碰撞	157
8.1.3 空间分割	159
8.2 物体碰撞	159
8.2.1 两个球体之间的碰撞	159
8.2.2 球体与长方体的碰撞	162
8.2.3 长方体之间的碰撞	167
8.3 本章小结	171
习题.....	172
第 9 章 多边形技术	173
9.1 深度值修正	174
9.1.1 修改投影矩阵	174
9.1.2 修正值的选择	175
9.1.3 应用举例	176
9.2 贴图	177

9.2.1 贴图网格	177
9.2.2 多边形裁剪	178
9.3 公告牌技术	179
9.3.1 无约束四边形	180
9.3.2 约束四边形	182
9.3.3 四边形带	182
9.4 模板阴影	183
9.4.1 构造边表	184
9.4.2 阴影渲染	188
9.4.3 实现	188
9.5 多边形简化	189
9.6 本章小结	193
习题	194
第 10 章 线性物理运动	195
10.1 位置方程	196
10.2 二阶微分方程	197
10.2.1 齐次方程	197
10.2.2 非齐次方程	199
10.2.3 初始条件	201
10.3 抛物运动	202
10.4 阻尼运动	205
10.5 摩擦力	206
10.6 本章小结	208
习题	209
第 11 章 旋转物理运动	211
11.1 旋转环境	212
11.1.1 角速度	212
11.1.2 离心力	213
11.1.3 科里奥利力	214
11.2 刚体运动	215
11.2.1 质心	215
11.2.2 角动量和扭矩	217
11.2.3 惯性张量	217
11.2.4 主惯性轴	222
11.3 振动	224
11.3.1 弹性运动	224
11.3.2 钟摆运动	227

11.4 本章小结	228
习题	229
第 12 章 流体仿真	231
12.1 波方程	232
12.2 近似导数	234
12.3 计算表面位移	236
12.4 算法实现	238
12.5 本章小结	242
习题	242
附录 A 复数	243
附录 B 三角恒等式	247
附录 C 坐标系	253
附录 D 泰勒级数	257
附录 E 习题答案	261
参考文献	265



第1章 向 量



在所有 3D 游戏引擎中,向量是一个非常重要的概念。向量可以用来表示空间中的点,比如游戏中物体的位置或者三角网格的顶点。它也可以表示空间方向,比如摄像机的指向或者三角网格的平面法向。对于一个成功的 3D 编程人员来说,掌握如何进行向量运算是一种基本的技能要求。

1.1 向量的性质

假设读者对向量已经有了基本的了解,由于向量的使用贯穿了整本书,所以对向量的性质进行一个快速的回顾是很有益处的。尽管会涉及到较多抽象的定义,但大多情况下将向量限定为是由 n 个实数定义的,这里的 n 为 2,3 或 4。一个 n 维向量 \mathbf{V} 可以表示为

$$\mathbf{V} = \langle V_1, V_2, \dots, V_n \rangle \quad (1.1)$$

V_i 称为向量 \mathbf{V} 的分量。这里用上了数字下标,通常情况下分量会被标上它们所属的坐标轴的名称。例如,三维点 P 的分量可以表示为 P_x, P_y 和 P_z 。

式(1.1)中的向量 \mathbf{V} 也可以用一个 n 行单列的矩阵来表示,即

$$\mathbf{V} = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} \quad (1.2)$$

这里的列向量与式(1.1)中具有分量列表形式的向量都表示相同的意义。向量通常用列向量的形式表示。有些时候也需要用单行 n 列的矩阵形式表示向量,这时用相应的列矩阵的转置矩阵来表示为

$$\mathbf{V}^T = [V_1 \ V_2 \ \dots \ V_n] \quad (1.3)$$

将向量乘以一个系数可以得到一个新的向量,新向量的各分量之间仍然保持原来相对的比例。系数 a 和向量 \mathbf{V} 的乘积可以定义为

$$a\mathbf{V} = \mathbf{V} \langle aV_1, aV_2, \dots, aV_n \rangle \quad (1.4)$$

当 $a=1$ 时,用较为简单的符号 $-\mathbf{V}$ 来表示向量 \mathbf{V} 负值。

向量的加、减运算都在向量的对应分量上进行。这样对于给定的向量 \mathbf{P} 和 \mathbf{Q} ,可以将加运算 $\mathbf{P} + \mathbf{Q}$ 定义为

$$\mathbf{P} + \mathbf{Q} = \langle P_1 + Q_1, P_2 + Q_2, \dots, P_n + Q_n \rangle \quad (1.5)$$

两个向量的减运算表示为 $\mathbf{P} - \mathbf{Q}$,实际上可以用简单的加运算来表示: $\mathbf{P} + (-\mathbf{Q})$ 。

根据以上的定义,可以得到向量运算的一些基本性质。

【定理 1.1】 对于给定的任何两个系数 a 和 b ,以及任何三个向量 \mathbf{P} 、 \mathbf{Q} 和 \mathbf{R} ,存在下述运算规律:

- (1) $\mathbf{P} + \mathbf{Q} = \mathbf{Q} + \mathbf{P}$
- (2) $(\mathbf{P} + \mathbf{Q}) + \mathbf{R} = \mathbf{P} + (\mathbf{Q} + \mathbf{R})$
- (3) $(ab)\mathbf{P} = a(b\mathbf{P})$
- (4) $a(\mathbf{P} + \mathbf{Q}) = a\mathbf{P} + a\mathbf{Q}$
- (5) $(a + b)\mathbf{P} = a\mathbf{P} + b\mathbf{P}$

根据实数的加法结合律和加法交换律,通过直接运算可以证明上述运算规律。

一个 n 维向量 \mathbf{V} 的模是一个系数,记作 $\|\mathbf{V}\|$,由下面的公式得到:

$$\|\mathbf{V}\| = \sqrt{\sum_{i=1}^n V_i^2} \quad (1.6)$$

向量的模有时又称为向量的范数或长度。模为 1 的向量称作单位长度向量,或者简称为单位向量。设向量 \mathbf{V} 表示一个三维的点或方向,则公式(1.6)可以展开为

$$\|\mathbf{V}\| = \sqrt{V_x^2 + V_y^2 + V_z^2} \quad (1.7)$$

如果向量 \mathbf{V} 至少有一个非零的分量,可以通过乘以 $1/\|\mathbf{V}\|$ 将向量 \mathbf{V} 的模变为单位长度,该操作叫做规格化。规格化在 3D 图形中经常用到。需要注意,这里的“规格化”与术语“法向量”毫无关系,“法向量”指的是在曲面上某个点处垂直于曲面的向量。

由求模公式(1.6)可得定理 1.2。

【定理 1.2】 对于任意给定的系数 a ,以及任意的两个向量 \mathbf{P} 与 \mathbf{Q} ,有以下性质:

- (1) $\|\mathbf{P}\| \geq 0$
- (2) 当且仅当 $\mathbf{P} = \langle 0, 0, \dots, 0 \rangle$ 时, $\|\mathbf{P}\| = 0$
- (3) $\|a\mathbf{P}\| = |a| \cdot \|\mathbf{P}\|$
- (4) $\|\mathbf{P} + \mathbf{Q}\| \leq \|\mathbf{P}\| + \|\mathbf{Q}\|$

证明

(1) 公式(1.6)中的被开方数是一些平方数的和,因为平方数不小于零,所以可以证得该性质。

(2) 设 $\mathbf{P} = \langle 0, 0, \dots, 0 \rangle$,则公式(1.6)中的被开方数为零,所以 $\|\mathbf{P}\| = 0$ 。反之,假设 $\|\mathbf{P}\| = 0$,则 \mathbf{P} 的各个分量必须为零,否则公式(1.6)中的被开方数为非零的正数,这与 $\|\mathbf{P}\| = 0$ 的假设矛盾。

(3) 根据公式(1.6)可以得到

$$\begin{aligned} \|a\mathbf{P}\| &= \sqrt{\sum_{i=1}^n a^2 P_i^2} = \sqrt{a^2 \sum_{i=1}^n P_i^2} \\ &= |a| \sqrt{\sum_{i=1}^n P_i^2} \\ &= |a| \|\mathbf{P}\| \end{aligned} \quad (1.8)$$

(4) 这就是所谓的三角不等式。如果将 \mathbf{P} 和 \mathbf{Q} 看作是三角形的两个边,则可以给出该等式的几何证明。如图 1.1 所示,由 $\mathbf{P} + \mathbf{Q}$ 可以得到三角形的第三边,显然第三边的长度不可能大于另两边的长度之和。

在 1.2 节介绍完点积之后,可以给出三角不等式的代数证明。

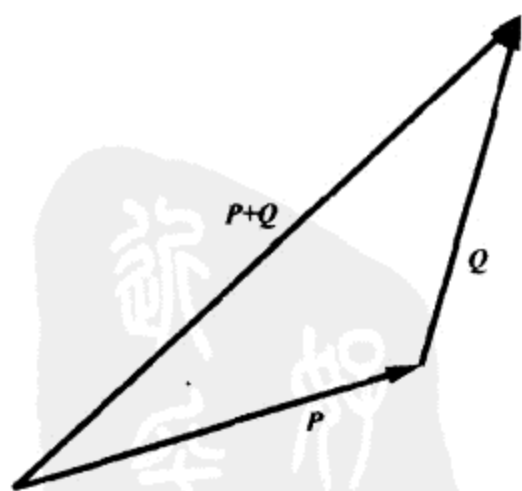


图 1.1 三角不等式

1.2 点 积

两个向量之间的点积有时又称做向量间的数量积或内积。在 3D 图形中经常大量地用到点积这种向量间的操作,这是因为点积为度量两个向量指向的差异提供了有效的手段。

【定理 1.3】 两个 n 维向量 P 和 Q 的点积记做 $P \cdot Q$,其结果由下面的公式给出,即

$$P \cdot Q = \sum_{i=1}^n P_i Q_i \quad (1.9)$$

该公式表明,两个向量的点积等于两个向量的每个对应分量的乘积之和。在三维空间中,可得

$$P \cdot Q = P_x Q_x + P_y Q_y + P_z Q_z \quad (1.10)$$

点积 $P \cdot Q$ 也可以用矩阵乘积的形式给出,即

$$P^T Q = [P_1 \ P_2 \ \cdots \ P_n] \begin{bmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_n \end{bmatrix} \quad (1.11)$$

其结果是一个 1×1 的矩阵(即一个数量),其唯一的分量等于公式(1.9)给出的和。

现在给出一个定理,这个定理揭示了点积普遍意义上的用途。

【定理 1.4】 对于给定的两个向量 P 和 Q ,点积 $P \cdot Q$ 满足下面的公式:

$$P \cdot Q = \|P\| \|Q\| \cos \alpha \quad (1.12)$$

这里, α 为连接 P 和 Q 的起始点和终点的线段之间的平面夹角。

证明 如图 1.2 所示, α 为线段 \overline{OP} 和 \overline{OQ} 之间的夹角。根据余弦定理(参看附录 B 的 B.6 部分),可得

$$\|P - Q\|^2 = \|P\|^2 + \|Q\|^2 - 2\|P\| \|Q\| \cos \alpha \quad (1.13)$$

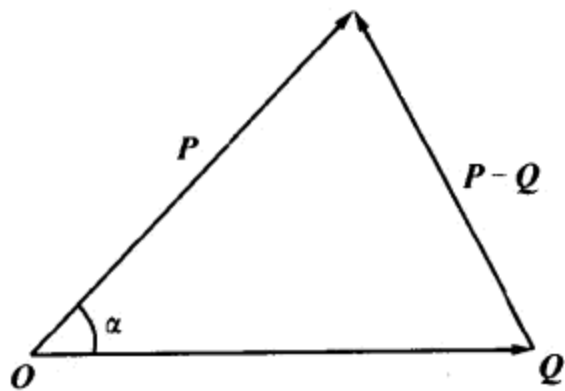
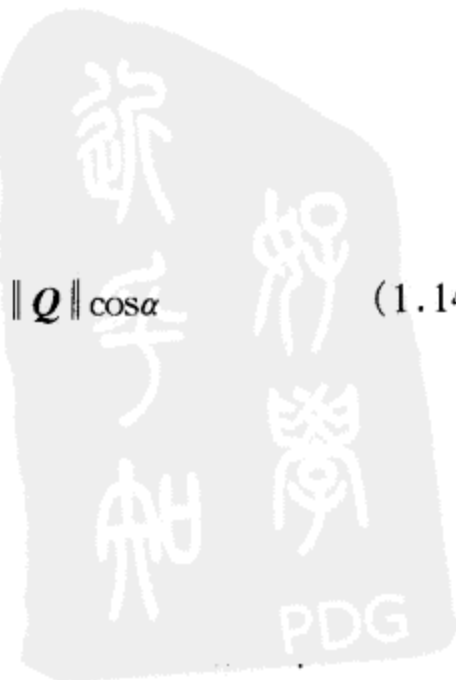


图 1.2 点积与两个向量的夹角有关

将公式展开得

$$\sum_{i=1}^n (P_i - Q_i)^2 = \sum_{i=1}^n P_i^2 + \sum_{i=1}^n Q_i^2 - 2\|P\| \|Q\| \cos \alpha \quad (1.14)$$

公式两边消去 P_i^2 项和 Q_i^2 项,可得



$$\sum_{i=1}^n -2P_i Q_i = -2\|P\|\|Q\|\cos\alpha \quad (1.15)$$

公式两边除以 -2 就可以得到证明结果。

从定理 1.4 可以直接得到两个结论。第一个结论是：当且仅当 $P \cdot Q = 0$ 时，向量 P 和 Q 相互垂直。这是因为当两个向量的夹角为 90° 时，余弦值为零。点积为 0 的向量称为正交向量。分量都为零的向量称为零向量，记作 $\mathbf{0} = \langle 0, 0, \dots, 0 \rangle$ ，零向量与任何向量都正交，因为对于任意向量 P ， $\mathbf{0} \cdot P$ 恒为零。

第二个结论是：点积的正负号说明两个向量指向同一方向的接近程度。如图 1.3 所示，假设平面通过向量 P 的起始点并且垂直于 P ，任何与 P 位于平面同一侧的向量与 P 的点积为正，任何与 P 位于平面不同侧的向量与 P 的点积为负。

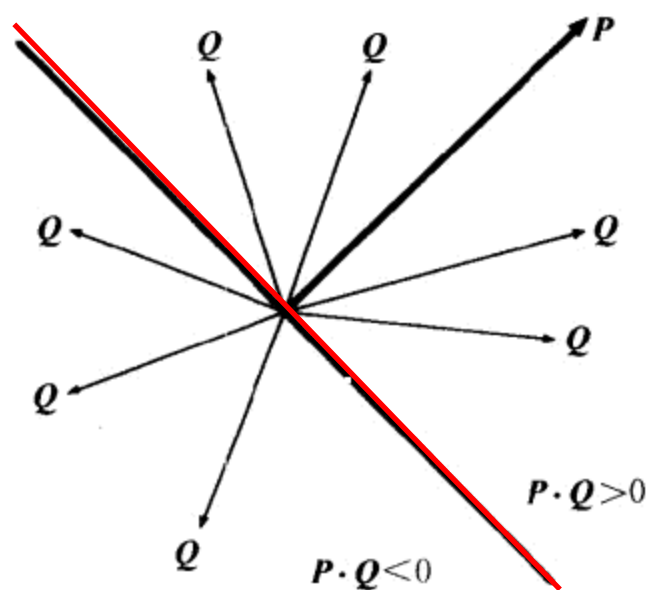


图 1.3 由点积可知两个向量是位于平面的同一侧还是不同侧

下面的定理给出了点积的其他一些性质。

【定理 1.5】 对于任意给定的系数 a 和任意三个向量 P , Q 和 R , 存在以下性质:

- (1) $P \cdot Q = Q \cdot P$
- (2) $(aP) \cdot Q = a(Q \cdot P)$
- (3) $P \cdot (Q + R) = P \cdot Q + P \cdot R$
- (4) $P \cdot P = \|P\|^2$
- (5) $|P \cdot Q| \leq \|P\|\|Q\|$

证明 根据实数的交换律和结合律的性质，很容易证得(1)、(2)和(3)。根据公式(1.6)对 $\|P\|$ 的定义和公式(1.9)对点积的定义，可以证得(4)。因为 $|\cos\alpha| \leq 1$ ，由定理 1.4 可以证得(5)。

用 P^2 来表示向量 P 与它自己的点积。根据定理 1.5 中的性质(4)，可以认为 $P \cdot P$, P^2 和 $\|P\|^2$ 表示相同的意义。

定理 1.5 中性质(5)就是众所周知的 Cauchy-Schwarz 不等式，根据它可以给出三角不等式的证明。

定理 1.2 中性质(4)(三角不等式)的证明。

从 $\|P + Q\|^2$ 出发，可以得到下面的计算过程：

$$\begin{aligned}
 \|P + Q\|^2 &= (P + Q) \cdot (P + Q) \\
 &= P^2 + Q^2 + 2P \cdot Q \\
 &\leq P^2 + Q^2 + 2\|P\|\|Q\| \\
 &= (\|P\| + \|Q\|)^2
 \end{aligned} \tag{1.16}$$

这里,在得到不等式的过程中用到了定理 1.5 中性质(5)。不等式两边取平方根,就可以得到证明结果。

在许多情况下,需要将一个向量 P 分解为平行于和垂直于另一个向量 Q 的分量。如图 1.4 所示,如果将向量 P 当作直角三角形的斜边,将向量 P 垂直投影到向量 Q 上,则可以得到 P 与 Q 之间夹角 α 的邻边。

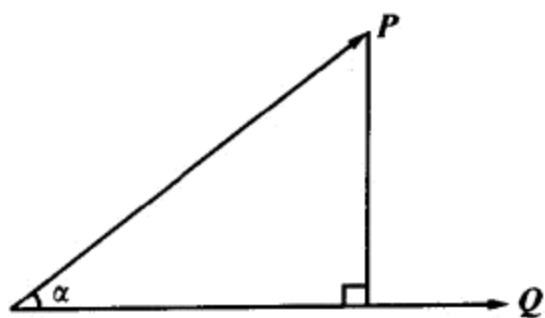


图 1.4 计算一个向量到另一个向量的投影

普通三角学告诉我们, α 的邻边等于 $\|P\| \cos \alpha$ 。定理 1.4 给出了一种不需知道 α 角而直接得到同样数值的计算方法

$$\|P\| \cos \alpha = \frac{P \cdot Q}{\|Q\|} \tag{1.17}$$

将得到的数值乘以单位向量 $Q/\|Q\|$,就可以得到具有该长度并且平行于 Q 的一个向量。

下面的公式表示向量 P 到向量 Q 的投影,可以记作 $\text{proj}_Q P$

$$\text{proj}_Q P = \frac{P \cdot Q}{\|Q\|^2} Q \tag{1.18}$$

向量 P 相对于向量 Q 的垂直分量记作 $\text{perp}_Q P$ 。如公式(1.19)所示,向量 P 减去平行分量就可以得到垂直分量为

$$\begin{aligned}
 \text{perp}_Q P &= P - \text{proj}_Q P \\
 &= P - \frac{P \cdot Q}{\|Q\|^2} Q
 \end{aligned} \tag{1.19}$$

P 到 Q 的投影是一个线性变换过程,所以可以将该过程表示为矩阵向量的乘积。在三维的情况下, $\text{proj}_Q P$ 也可以用下面的公式来计算,即

$$\text{proj}_Q P = \frac{1}{\|Q\|^2} \begin{bmatrix} Q_x^2 & Q_x Q_y & Q_x Q_z \\ Q_x Q_y & Q_y^2 & Q_y Q_z \\ Q_x Q_z & Q_y Q_z & Q_z^2 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \tag{1.20}$$

1.3 叉积

两个三维向量的叉积又称为它们的向量积,其结果是得到一个新的向量,新向量垂直于原来两个相乘的向量。在计算机图形学中经常用到叉积的这种性质,比如,已知曲面上一点处的两个不同的切向量,通过计算这两个切向量的叉积,就可以得到曲面在该点处的法向量。

【定义 1.6】 两个 3D 向量 \mathbf{P} 和 \mathbf{Q} 的叉积记作 $\mathbf{P} \times \mathbf{Q}$, 其结果向量为

$$\mathbf{P} \times \mathbf{Q} = \langle P_y Q_z - P_z Q_y, P_z Q_x - P_x Q_z, P_x Q_y - P_y Q_x \rangle \quad (1.21)$$

为了方便记忆该公式,常用的工具是伪行列式,通过计算伪行列式可以计算得到叉积

$$\mathbf{P} \times \mathbf{Q} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ P_x & P_y & P_z \\ Q_x & Q_y & Q_z \end{vmatrix} \quad (1.22)$$

这里, \mathbf{i} , \mathbf{j} 和 \mathbf{k} 是分别平行于 x 轴、 y 轴、 z 轴的单位向量

$$\begin{aligned} \mathbf{i} &= \langle 1, 0, 0 \rangle \\ \mathbf{j} &= \langle 0, 1, 0 \rangle \\ \mathbf{k} &= \langle 0, 0, 1 \rangle \end{aligned} \quad (1.23)$$

公式(1.22)的右边部分称为伪行列式,这是因为矩阵的第一行包含了向量,而其他部分则是系数。尽管如此,按照一般的行列式计算方法进行计算能够得到正确的叉积结果,如下面的公式所示:

$$\begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ P_x & P_y & P_z \\ Q_x & Q_y & Q_z \end{vmatrix} = \mathbf{i}(P_y Q_z - P_z Q_y) - \mathbf{j}(P_x Q_z - P_z Q_x) + \mathbf{k}(P_x Q_y - P_y Q_x) \quad (1.24)$$

叉积 $\mathbf{P} \times \mathbf{Q}$ 也可以用线性变换的形式表示,变换矩阵派生于 \mathbf{P} , 作用于 \mathbf{Q} , 如下所示:

$$\mathbf{P} \times \mathbf{Q} = \begin{bmatrix} 0 & -P_z & P_y \\ P_z & 0 & -P_x \\ -Q_y & P_x & 0 \end{bmatrix} \begin{bmatrix} Q_x \\ Q_y \\ Q_z \end{bmatrix} \quad (1.25)$$

前面已讲过,叉积 $\mathbf{P} \times \mathbf{Q}$ 得到一个与 \mathbf{P} 和 \mathbf{Q} 都垂直的一个向量。这一点可以用下面的定理来概括。

【定理 1.7】 设 \mathbf{P} 和 \mathbf{Q} 为任意的两个 3D 向量,则有 $(\mathbf{P} \times \mathbf{Q}) \cdot \mathbf{P} = 0$ 以及 $(\mathbf{P} \times \mathbf{Q}) \cdot \mathbf{Q} = 0$ 。

证明 根据叉积和点积的定义,可以得到以下推导过程:

$$\begin{aligned} (\mathbf{P} \times \mathbf{Q}) \cdot \mathbf{P} &= \langle P_y Q_z - P_z Q_y, P_z Q_x - P_x Q_z, P_x Q_y - P_y Q_x \rangle \cdot \mathbf{P} \\ &= P_x P_y Q_z - P_x P_z Q_y + P_y P_z Q_x - P_x P_y Q_z + P_x P_z Q_y - P_y P_z Q_x \\ &= 0 \end{aligned} \quad (1.26)$$

同理可以证明 $(\mathbf{P} \times \mathbf{Q}) \cdot \mathbf{Q} = 0$ 。

参照前面的推导过程可知,对于任意给定的三个 3D 向量 \mathbf{P} 、 \mathbf{Q} 和 \mathbf{R} , 等式 $(\mathbf{P} \times \mathbf{Q}) \cdot \mathbf{R}$ 的值可以通过计算下面的行列式得到:

$$(\mathbf{P} \times \mathbf{Q}) \cdot \mathbf{R} = \begin{vmatrix} P_x & P_y & P_z \\ Q_x & Q_y & Q_z \\ R_x & R_y & R_z \end{vmatrix} \quad (1.27)$$

如果向量 \mathbf{P} 、 \mathbf{Q} 和 \mathbf{R} 中的任何一个可以表示为其他的两个向量的线性组合,则行列式的值为零。这包括 $\mathbf{R} = \mathbf{P}$ 或 $\mathbf{R} = \mathbf{Q}$ 的情况。

与点积类似,叉积也具有三角学意义。

【定理 1.8】 对于给定的两个 3D 向量 \mathbf{P} 和 \mathbf{Q} ,它们的叉积 $\mathbf{P} \times \mathbf{Q}$ 满足下面的公式

$$\|\mathbf{P} \times \mathbf{Q}\| = \|\mathbf{P}\| \|\mathbf{Q}\| \sin \alpha, \quad (1.28)$$

这里的 α 为连接 \mathbf{P} 和 \mathbf{Q} 的起始点与终点的两条线段之间的夹角。

证明 取 $\|\mathbf{P} \times \mathbf{Q}\|^2$ 的平方,可得

$$\begin{aligned} \|\mathbf{P} \times \mathbf{Q}\|^2 &= \|\langle P_y Q_z - P_z Q_y, P_z Q_x - P_x Q_z, P_x Q_y - P_y Q_x \rangle\|^2 \\ &= (P_y Q_z - P_z Q_y)^2 + (P_z Q_x - P_x Q_z)^2 + (P_x Q_y - P_y Q_x)^2 \\ &= (P_y^2 + P_z^2) Q_x^2 + (P_x^2 + P_z^2) Q_y^2 + (P_x^2 + P_y^2) Q_z^2 - \\ &\quad 2P_x Q_x P_y Q_y - 2P_x Q_x P_z Q_z - 2P_y Q_y P_z Q_z \end{aligned} \quad (1.29)$$

在公式的右边加上并减去 $P_x^2 Q_x^2 + P_y^2 Q_y^2 + P_z^2 Q_z^2$,可得

$$\begin{aligned} \|\mathbf{P} \times \mathbf{Q}\|^2 &= (P_x^2 + P_y^2 + P_z^2)(Q_x^2 + Q_y^2 + Q_z^2) - (P_x Q_x + P_y Q_y + P_z Q_z)^2 \\ &= \|\mathbf{P}\|^2 \|\mathbf{Q}\|^2 - (\mathbf{P} \cdot \mathbf{Q})^2 \end{aligned} \quad (1.30)$$

用公式(1.12)取代上面公式中的点积,可得

$$\begin{aligned} \|\mathbf{P} \times \mathbf{Q}\|^2 &= \|\mathbf{P}\|^2 \|\mathbf{Q}\|^2 - \|\mathbf{P}\|^2 \|\mathbf{Q}\|^2 \cos^2 \alpha \\ &= \|\mathbf{P}\|^2 \|\mathbf{Q}\|^2 (1 - \cos^2 \alpha) \\ &= \|\mathbf{P}\|^2 \|\mathbf{Q}\|^2 \sin^2 \alpha \end{aligned} \quad (1.31)$$

公式两边开平方,即可证得定理 1.8。

如图 1.5 所示,由定理 1.8 可知叉积 $\mathbf{P} \times \mathbf{Q}$ 的模等于由向量 \mathbf{P} 和 \mathbf{Q} 组成的平行四边形的面积。因此,对于顶点为 \mathbf{V}_1 、 \mathbf{V}_2 和 \mathbf{V}_3 的任意一个三角形,利用下面的公式可以计算其面积 A ,即

$$A = \frac{1}{2} \|(\mathbf{V}_2 - \mathbf{V}_1) \times (\mathbf{V}_3 - \mathbf{V}_1)\| \quad (1.32)$$

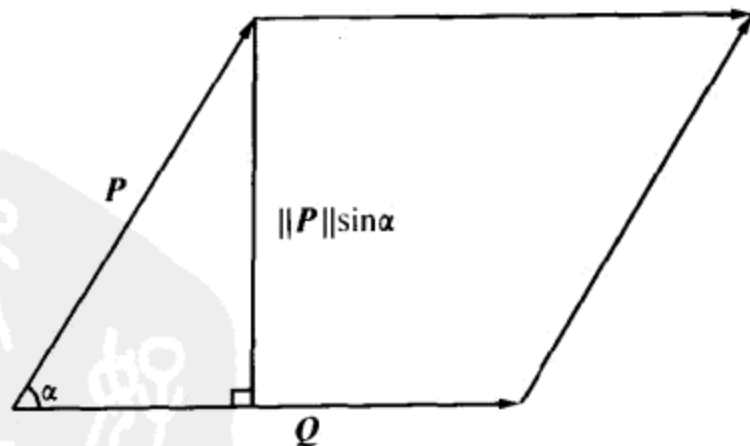


图 1.5 两个向量叉积的模

我们知道,叉积得到的非零向量一定垂直于进行叉积运算的两个向量,但满足该要求的向量的方向有两种。为了确定叉积的方向,这里引入了叉积遵循的右手法则。如图 1.6 所示,如果右手手指顺着向量 P 的方向,而手掌面向向量 Q 的方向,那么大拇指所指的方向就是叉积 $P \times Q$ 的方向。

这里假设有三个单位向量 i, j 和 k ,它们分别指向 x 轴、 y 轴和 z 轴的正向。如果将这三个向量按照圆形循环的方式排序,即: i 在 j 之前, j 在 k 之前, k 又在 i 前,则顺序相接的两个向量的叉积等于第三个向量

$$\begin{aligned} i \times j &= k \\ j \times k &= i \\ k \times i &= j \end{aligned} \quad (1.33)$$

按照相反顺序相接的两个向量的叉积等于第三个向量的负值,如下所示:

$$\begin{aligned} j \times i &= -k \\ k \times j &= -i \\ i \times k &= -j \end{aligned} \quad (1.34)$$

叉积的其他一些性质由下面的定理给出。

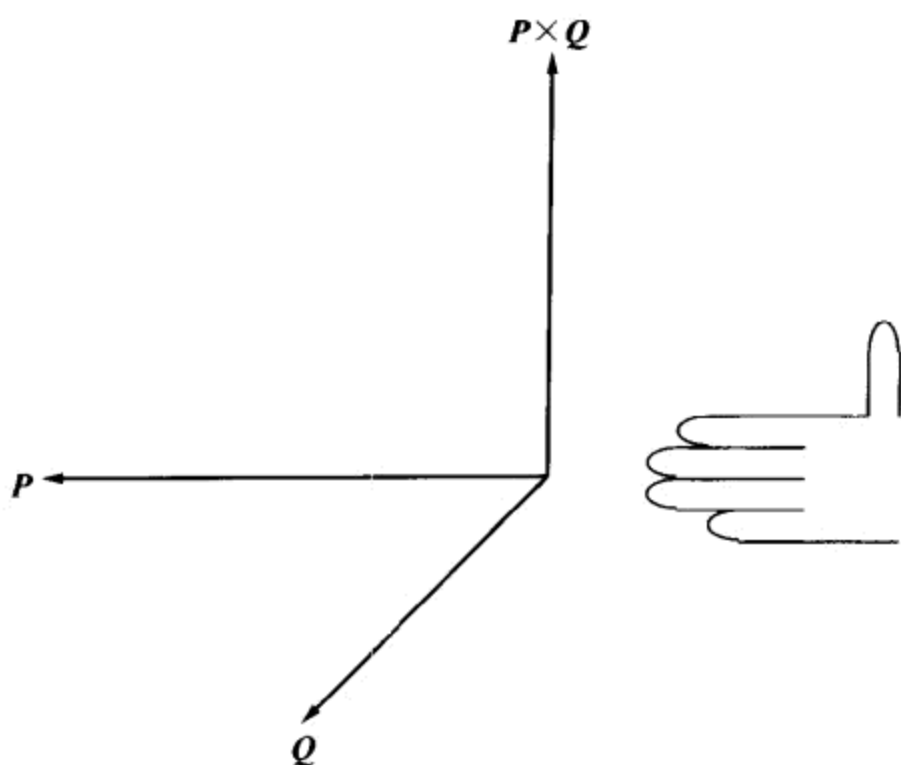


图 1.6 右手法则用于确定叉积的方向

【定理 1.9】 对于给定的两个系数 a 和 b 和任意三个 3D 向量 P, Q 和 R ,有以下性质存在:

- (1) $Q \times P = -(P \times Q)$
- (2) $(aP) \times Q = a(P \times Q)$
- (3) $P \times (Q + R) = P \times Q + P \times R$
- (4) $P \times P = \mathbf{0} = \langle 0, 0, 0 \rangle$
- (5) $(P \times Q) \cdot R = (R \times P) \cdot Q = (Q \times R) \cdot P$
- (6) $P \times (Q \times P) = P \times Q \times R = P^2 Q - (P \cdot Q)P$

证明 从叉积的定义以及实数的结合律和交换律可以直接得到性质(1)~(4)。性质(5)可以直接从公式(1.27)证明。对于性质(6),可以直接进行推导

$$\begin{aligned} \mathbf{P} \times (\mathbf{Q} \times \mathbf{R}) &= \mathbf{P} \times [-(\mathbf{P} \times \mathbf{Q})] \\ &= -[-(\mathbf{P} \times \mathbf{Q}) \times \mathbf{P}] \\ &= \mathbf{P} \times \mathbf{Q} \times \mathbf{R} \end{aligned} \quad (1.35)$$

直接对 x 分量进行计算可得

$$\begin{aligned} (\mathbf{P} \times \mathbf{Q} \times \mathbf{R})_x &= (\langle P_y Q_z - P_z Q_y, P_z Q_x - P_x Q_z, P_x Q_y - P_y Q_x \rangle \times \mathbf{P}) \\ &= (P_z Q_x - P_x Q_z)P_x - (P_x Q_y - P_y Q_x)P_y \\ &= (P_y^2 + P_z^2)Q_x - (P_y Q_y + P_z Q_z)P_x \end{aligned} \quad (1.36)$$

在公式中加上并减去 $P_x^2 Q_x$ 项,可得

$$\begin{aligned} &(P_y^2 + P_z^2)Q_x - (P_y Q_y + P_z Q_z)P_x \\ &= (P_y^2 + P_z^2)Q_x + P_x^2 Q_x - (P_y Q_y + P_z Q_z)P_x - P_x^2 Q_x \\ &= (P_x^2 + P_y^2 + P_z^2)Q_x - (P_x Q_x + P_y Q_y + P_z Q_z)P_x \\ &= P^2 Q_x - (\mathbf{P} \cdot \mathbf{Q})P_x \end{aligned} \quad (1.37)$$

用同样的方式可以得到 y 分量和 z 分量的情况。

从定理 1.9 的性质(1)可以看出,叉积不满足简单的交换操作,如果交换了叉积中向量的顺序会得到一个负的结果,也就是说叉积是不满足交换律的。另外需要注意的是,叉积是不满足结合律的,对于任意给定的三个向量 \mathbf{P} 、 \mathbf{Q} 和 \mathbf{R} , $(\mathbf{P} \times \mathbf{Q}) \times \mathbf{R} \neq \mathbf{P} \times (\mathbf{Q} \times \mathbf{R})$ 。

例如,设 $\mathbf{P} = \langle 1, 1, 0 \rangle$, $\mathbf{Q} = \langle 0, 1, 1 \rangle$, $\mathbf{R} = \langle 1, 0, 1 \rangle$,先计算 $(\mathbf{P} \times \mathbf{Q}) \times \mathbf{R}$

$$\begin{aligned} \mathbf{P} \times \mathbf{Q} &= \begin{vmatrix} i & j & k \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{vmatrix} = \langle 1, -1, 1 \rangle \\ \mathbf{P} \times \mathbf{Q} \times \mathbf{R} &= \begin{vmatrix} i & j & k \\ 1 & -1 & 1 \\ 1 & 0 & 1 \end{vmatrix} = \langle -1, 0, 1 \rangle \end{aligned} \quad (1.38)$$

下面计算 $\mathbf{P} \times (\mathbf{Q} \times \mathbf{R})$

$$\begin{aligned} \mathbf{Q} \times \mathbf{R} &= \begin{vmatrix} i & j & k \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{vmatrix} = \langle 1, 1, -1 \rangle \\ \mathbf{P} \times (\mathbf{Q} \times \mathbf{R}) &= \begin{vmatrix} i & j & k \\ 1 & 1 & 0 \\ 1 & 1 & -1 \end{vmatrix} = \langle -1, 1, 0 \rangle \end{aligned} \quad (1.39)$$

通过比较可以看出, $(\mathbf{P} \times \mathbf{Q}) \times \mathbf{R}$ 和 $\mathbf{P} \times (\mathbf{Q} \times \mathbf{R})$ 具有不同的结果。

1.4 向量空间

到目前为止,我们讨论的向量都属于一种集合,这种集合称为向量空间。对向量空间进行讨论将有助于一些概念的介绍,这些概念对于下一章矩阵的讨论非常重要。

【定理 1.10】 向量空间是一个集合 V , 它的元素称为向量。这里定义向量空间中的加法运算和乘数运算, 并介绍向量空间的一些性质。

- (1) V 对于加法运算封闭。也就是说, 对于 V 中的任意向量 P 和 Q , 它们的和 $P + Q$ 仍然是 V 的元素。
- (2) V 对于乘数运算封闭。也就是说, 对于任意的实数 a 和 V 中的任意向量 P , 乘积 aP 仍然是 V 的元素。
- (3) 向量空间 V 中存在一个元素 0 , 对于 V 中的任意元素 P , $P + 0 = 0 + P = P$ 。
- (4) 对于向量空间 V 中的任意元素 P , 在向量空间 V 中存在对应的向量 Q , 使得 $P + Q = 0$ 。
- (5) 加法满足结合律。也就是说, 对于 V 中任意的向量 P , Q 和 R , $(P + Q) + R = P + (Q + R)$ 。
- (6) 乘数操作满足结合律。对于任意的实数 a 和 b , 以及向量空间 V 中的元素 P , $(ab)P = a(bP)$ 。
- (7) 乘数操作对于向量加法满足分配率。对于任意的实数 a 和向量空间 V 中的任意元素 P 和 Q , $a(P + Q) = aP + aQ$ 。
- (8) 系数加法对于乘数操作满足分配率。对于任意的实数 a 和 b 以及 V 中的任意向量 P , $(a + b)P = aP + bP$ 。

向量空间的许多性质在 1.1 节已经提到, 这些性质对于具有 n 元实数组的向量来说很容易证明。将由 n 元数组组成的向量空间记作 R^n 。比如, 包含所有 3D 向量的向量空间记作 R^3 。

每一个向量空间都可以由一组向量的线性组合来生成, 这些向量的组合称为向量组, 生成向量空间的向量组称为向量空间的基。在详细解释基之前, 这里先介绍向量间的线性无关。

【定义 1.11】 对于一组向量 $\{e_1, e_2, \dots, e_n\}$, 如果不存在这样一组实数 a_1, a_2, \dots, a_n , 其中至少有一个 a_i 不为零, 使得下式成立

$$a_1 e_1 + a_2 e_2 + \dots + a_n e_n = 0 \quad (1.40)$$

则称这组向量线性无关, 否则称这组向量线性相关。

一个 n 维向量空间可以由 n 个线性无关的向量组成的向量集合生成, 这样的生成向量集合称为基。基的正式定义如定义 1.12。

【定义 1.12】 向量空间 V 的基 B 是一个由 n 个线性无关的向量组成的集合, 即 $B = \{e_1, e_2, \dots, e_n\}$, 对于向量空间 V 中的任意一个元素 P , 都存在一组实数使得

$$P = a_1 e_1 + a_2 e_2 + \dots + a_n e_n \quad (1.41)$$

n 维向量空间的每个基都具有 n 个向量。比如, 在 R^3 中不可能找到包含 4 个线性无关的向量的集合, 并且用两个线性无关的向量也不可能生成整个向量空间。

任意向量空间 R^n 都具有无限个基, 这里对基的形式进行一些限定, 以使基具有特定的性质。

【定义 1.13】 对于向量空间的基 $B = \{e_1, e_2, \dots, e_n\}$, 如果对于任意的 $i \neq j$ 都有 $e_i \cdot e_j = 0$, 则称该基为正交基。

实际上, 两个向量的点积为零已经隐含了这两个向量是线性无关的。

【定理 1.14】 对于给定的两个向量 e_1 和 e_2 , 如果 $e_1 \cdot e_2 = 0$, 则 e_1 和 e_2 线性无关。

证明 用反证法来证明 e_1 和 e_2 线性无关。假设 e_1 和 e_2 线性相关, 则存在系数 a_1 和 a_2 使得 $a_1 e_1 + a_2 e_2 = 0$, 按照假设 a_1 和 a_2 是不能为零的, 则有 $e_2 = -\frac{a_1}{a_2} e_1$, 由此可得 $e_1 \cdot e_2 = -\frac{a_1}{a_2} e_1^2 \neq 0$, 这与 $e_1 \cdot e_2 = 0$ 相矛盾, 故定理得证。

对基的另一个形式限定是基的所有向量都具有单位长度。为了方便表示, 这里引入 Kronecker delta 符号 δ_{ij} , 其定义如下

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad (1.42)$$

【定理 1.15】 如果对于每一对 (i, j) 有 $e_1 \cdot e_2 = \delta_{ij}$, 则基 $\mathcal{B} = \{e_1, e_2, \dots, e_n\}$ 称为向量空间的正交规范基。

显然向量组 $\{i, j, k\}$ 是 R^3 的一个正交规范基。例如三个向量 $\langle \frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0 \rangle, \langle -\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0 \rangle$ 和 $\langle 0, 0, 1 \rangle$ 组成了 R^3 的一个正交规范基。

这里有一个方法可以将具有 n 个线性无关向量的向量组转换为向量空间 R^n 的正交基。基本思想是减去向量在前面每个向量上的投影, 不管结果如何, 它肯定与前面所有的向量正交。详细过程在算法 1.16 中进行了描述。

【算法 1.16】 Gram-Schmidt 正交规范化: 对于由 n 个线性无关向量组成的向量组 $\mathcal{B} = \{e_1, e_2, \dots, e_n\}$, 该算法产生向量组 $\mathcal{B}' = \{e'_1, e'_2, \dots, e'_n\}$, 当 $i \neq j$ 时, $e'_i \cdot e'_j = 0$ 。

(1) 设 $e'_1 = e_1$;

(2) 从 $i=2$ 开始;

(3) 从 e_i 中减去 e_i 在向量 $e'_1, e'_2, \dots, e'_{i-1}$ 上的投影, 结果为 e'_i , 即

$$e'_i = e_i - \sum_{k=1}^{i-1} \frac{e_i \cdot e'_k}{e'^2_k} e'_k \quad (1.43)$$

(4) 如果 $i < n$, i 加 1, 并跳到步骤(3)。

1.5 本章小结

点积

两个 n 维向量 P 和 Q 的点积记做 $P \cdot Q$, 其数量结果为

$$P \cdot Q = \sum_{i=1}^n P_i Q_i = P_1 Q_1 + P_2 Q_2 + \dots + P_n Q_n$$

点积与向量 P 和 Q 之间的夹角 α 有关, 点积的另一个计算公式为

$$P \cdot Q = \|P\| \|Q\| \cos \alpha$$

向量投影

向量 P 到向量 Q 的投影为

$$\text{proj}_Q \mathbf{P} = \frac{\mathbf{P} \cdot \mathbf{Q}}{\|\mathbf{Q}\|^2} \mathbf{Q}$$

向量 \mathbf{P} 相对于向量 \mathbf{Q} 的垂直分量为

$$\text{perp}_Q \mathbf{P} = \mathbf{P} - \text{proj}_Q \mathbf{P} = \mathbf{P} - \frac{\mathbf{P} \cdot \mathbf{Q}}{\|\mathbf{Q}\|^2} \mathbf{Q}$$

叉积

两个 3D 向量 \mathbf{P} 和 \mathbf{Q} 的叉积记作 $\mathbf{P} \times \mathbf{Q}$, 其结果向量为

$$\mathbf{P} \times \mathbf{Q} = \langle P_y Q_z - P_z Q_y, P_z Q_x - P_x Q_z, P_x Q_y - P_y Q_x \rangle$$

叉积也可以写成向量矩阵乘积的形式

$$\mathbf{P} \times \mathbf{Q} = \begin{bmatrix} 0 & -P_z & P_y \\ P_z & 0 & -P_x \\ -P_y & P_x & 0 \end{bmatrix} \begin{bmatrix} Q_x \\ Q_y \\ Q_z \end{bmatrix}$$

叉积的模与 \mathbf{P} 和 \mathbf{Q} 之间的夹角 α 有关, 下面的公式表明了这种关系

$$\|\mathbf{P} \times \mathbf{Q}\| = \|\mathbf{P}\| \|\mathbf{Q}\| \sin \alpha$$

Gram-Schmidt 正交规范化

通过对 n 维向量空间的基 $\mathcal{B} = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n\}$ 进行正交规范化, 可以得到一个新的基 $\mathcal{B}' = \{\mathbf{e}'_1, \mathbf{e}'_2, \dots, \mathbf{e}'_n\}$, 正交规范化用到下面公式

$$\mathbf{e}'_i = \mathbf{e}_i - \sum_{k=1}^{i-1} \frac{\mathbf{e}_i \cdot \mathbf{e}_k}{\|\mathbf{e}_k\|^2} \mathbf{e}_k$$

习 题

1. 设 $\mathbf{P} = \langle 2, 2, 1 \rangle$, $\mathbf{Q} = \langle 1, -2, 0 \rangle$, 计算

(1) $\mathbf{P} \cdot \mathbf{Q}$

(2) $\mathbf{P} \times \mathbf{Q}$

(3) $\text{proj}_Q \mathbf{P}$

2. 正交规范化下面的向量组:

$$\mathbf{e}_1 = \left\langle \frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0 \right\rangle \quad \mathbf{e}_2 = \langle -1, 1, -1 \rangle$$

$$\mathbf{e}_3 = \langle 0, -2, -2 \rangle$$

3. 证明: 对于任意的三个 3D 向量 \mathbf{P} , \mathbf{Q} 和 \mathbf{R} , 下式成立。

$$\mathbf{P} \times \mathbf{Q} \times \mathbf{R} = (\mathbf{P} \cdot \mathbf{R}) \mathbf{Q} - (\mathbf{Q} \cdot \mathbf{R}) \mathbf{P}$$

4. 用 C++ 类实现对 3D 向量的封装。该类应该拥有表示向量的 x, y, z 三个分量的浮点形式的成员数据。该类除了拥有不做任何操作的默认构造函数外, 还应该具有入口参数为三个浮点数的构造函数, 该构造函数根据入口参数对向量分量进行赋值。该类对向量的加、减、乘以及除以数量系数的运算符进行重载, 并重载点积和叉积。最后, 编制一个计算 3D 向量对象的模的函数。

第 2 章 矩 阵

$$\mathbf{F} + \mathbf{G} = \begin{bmatrix} F_{11} + G_{11} & F_{12} + G_{12} & \cdots & F_{1m} + G_{1m} \\ F_{21} + G_{21} & F_{22} + G_{22} & \cdots & F_{2m} + G_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ F_{n1} + G_{n1} & F_{n2} + G_{n2} & \cdots & F_{nm} + G_{nm} \end{bmatrix}$$



第 1 章中,已经几次提到了矩阵的概念。在进一步讨论矩阵在 3D 游戏编程中的应用之前,需要对矩阵的性质有更正确的认识。

2.1 矩阵的性质

一个 $n \times m$ 矩阵 M 就是一个有 n 行和 m 列的数组。如果 $n = m$, 则矩阵 M 为方阵, 把矩阵 M 第 j 列的第 i 个元素记作 M_{ij} 。例如, 若 F 是一个 3×4 的矩阵, 那么它的表达式为

$$F = \begin{bmatrix} F_{11} & F_{12} & F_{13} & F_{14} \\ F_{21} & F_{22} & F_{23} & F_{24} \\ F_{31} & F_{32} & F_{33} & F_{34} \end{bmatrix} \quad (2.1)$$

其中 $i = j$ 的元素叫做矩阵的主对角线元素。只有主对角线上的元素不为 0 的方阵叫做对角阵。

一个 $n \times m$ 矩阵 M 的转置, 记做 M^T , 是一个 $m \times n$ 矩阵, 其中 $M_{ij}^T = M_{ji}$ 。上式 (2.1) 中矩阵 F 的转置是

$$F^T = \begin{bmatrix} F_{11} & F_{21} & F_{31} \\ F_{12} & F_{22} & F_{32} \\ F_{13} & F_{23} & F_{33} \\ F_{14} & F_{24} & F_{34} \end{bmatrix} \quad (2.2)$$

与向量(其实是 $n \times 1$ 矩阵)一样, 矩阵也定义了标量乘积。假设 a 是一个标量系数, M 是 $n \times m$ 矩阵, 那么 aM 就是

$$aM = Ma = \begin{bmatrix} aM_{11} & aM_{12} & \cdots & aM_{1m} \\ aM_{21} & aM_{22} & \cdots & aM_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ aM_{n1} & aM_{n2} & \cdots & aM_{nm} \end{bmatrix} \quad (2.3)$$

与向量类似, 矩阵也有加法运算。假设 F 和 G 都是 $n \times m$ 的矩阵, 则 $F + G$ 就是

$$F + G = \begin{bmatrix} F_{11} + G_{11} & F_{12} + G_{12} & \cdots & F_{1m} + G_{1m} \\ F_{21} + G_{21} & F_{22} + G_{22} & \cdots & F_{2m} + G_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ F_{n1} + G_{n1} & F_{n2} + G_{n2} & \cdots & F_{nm} + G_{nm} \end{bmatrix} \quad (2.4)$$

如果矩阵 F 的列数和矩阵 G 的行数相等, 则矩阵 F 和 M 就可以相乘。假设 F 是 $n \times m$ 的矩阵, G 是 $m \times p$ 的矩阵, 那么 FG 就是 $n \times p$ 的矩阵, 它在 (i, j) 位置上的元素可表示为

$$(FG)_{ij} = \sum_{k=1}^m F_{ik}G_{kj} \quad (2.5)$$

其实也可以把矩阵 FG 在位置 (i, j) 上的元素看作是矩阵 F 的第 i 行和矩阵 G 的第 j 列元素之间的点积。

单位阵是 $n \times n$ 的矩阵, 记作 I_n , 对于 $n \times n$ 的矩阵 M , 则有 $MI_n = I_nM = M$ 。单位阵可表示为以下形式

$$\mathbf{I}_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \quad (2.6)$$

因为单位阵的尺寸可由具体应用情况得出。通常把下标 n 去掉, 而把单位阵记做 \mathbf{I} , 以下两个定理给出了矩阵的几个性质。

【定理 2.1】 任意给出两个标量系数 a, b 以及三个 $n \times m$ 的矩阵 $\mathbf{F}, \mathbf{G}, \mathbf{H}$, 则以下等式成立:

- (1) $\mathbf{F} + \mathbf{G} = \mathbf{G} + \mathbf{F}$
- (2) $(\mathbf{F} + \mathbf{G}) + \mathbf{H} = \mathbf{F} + (\mathbf{G} + \mathbf{H})$
- (3) $a(b\mathbf{F}) = (ab)\mathbf{F}$
- (4) $a(\mathbf{F} + \mathbf{G}) = a\mathbf{F} + a\mathbf{G}$
- (5) $(a + b)\mathbf{F} = a\mathbf{F} + b\mathbf{F}$

与向量一样, 这些性质可由实数的交换律和结合律轻松得到证明。

【定理 2.2】 对于任意的标量系数 a , $n \times m$ 矩阵 \mathbf{F} , $m \times p$ 矩阵 \mathbf{G} , $p \times q$ 矩阵 \mathbf{H} , 有以下性质:

- (1) $(a\mathbf{F})\mathbf{G} = a(\mathbf{F}\mathbf{G})$
- (2) $(\mathbf{F}\mathbf{G})\mathbf{H} = \mathbf{F}(\mathbf{G}\mathbf{H})$
- (3) $(\mathbf{F}\mathbf{G})^T = \mathbf{G}^T\mathbf{F}^T$

证明 (1) 由式(2.5)给出的矩阵乘法的定义, 可得矩阵 $(a\mathbf{F})\mathbf{G}$ 的 (i, j) 元素为

$$\begin{aligned} [(a\mathbf{F})\mathbf{G}]_{ij} &= \sum_{k=1}^m (a\mathbf{F})_{ik}G_{kj} \\ &= \sum_{k=1}^m a(F_{ik}G_{kj}) \\ &= a \sum_{k=1}^m F_{ik}G_{kj} \\ &= a(\mathbf{F}\mathbf{G})_{ij} \end{aligned} \quad (2.7)$$

(2) 同样利用式(2.5)可得矩阵 $(\mathbf{F}\mathbf{G})\mathbf{H}$ 的 (i, j) 元素为

$$\begin{aligned} [(\mathbf{F}\mathbf{G})\mathbf{H}]_{ij} &= \sum_{k=1}^p (\mathbf{F}\mathbf{G})_{ik}H_{kj} \\ &= \sum_{k=1}^p \left(\sum_{l=1}^m F_{il}G_{lk} \right) H_{kj} \\ &= \sum_{l=1}^m F_{il} \left(\sum_{k=1}^p G_{lk}H_{kj} \right) \\ &= \sum_{l=1}^m F_{il}(\mathbf{G}\mathbf{H})_{lj} \\ &= [\mathbf{F}(\mathbf{G}\mathbf{H})]_{ij} \end{aligned} \quad (2.8)$$

(3) 利用式(2.5), 并在应用或取消每一个转置操作时把相应的序号反转, 就可得到位于

(i, j) 的矩阵 $(FG)^T$ 的元素为

$$\begin{aligned} (FG)^T_{ij} &= (FG)_{ji} \\ &= \sum_{k=1}^m F_{jk} G_{ki} \\ &= \sum_{k=1}^m F_{kj}^T G_{ik}^T \\ &= (G^T F^T)_{ij} \end{aligned} \quad (2.9)$$

2.2 线性方程组

矩阵提供了一种有效并且简便的线性方程组的表示方法。例如,由以下方程给出的线性方程组

$$\begin{aligned} 3x + 2y - 3z &= 5 \\ 4x - 3y + 6z &= 1 \\ x - z &= 3 \end{aligned} \quad (2.10)$$

可以用以下的矩阵形式表示

$$\begin{bmatrix} 3 & 2 & -3 \\ 4 & -3 & 6 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \\ 3 \end{bmatrix} \quad (2.11)$$

在未知向量 $\langle x, y, z \rangle$ 前面的矩阵叫做系数矩阵,在等号右边的列向量叫做常数向量。常数向量不为 0(如上例)的线性方程组叫做非齐次方程组,而常数向量为 0 的线性方程组叫齐次方程组。

把系数矩阵和常数向量并置在一起,然后对其执行基本行变换,就可以得到线性方程组的解。

【定义 2.3】 所谓基本行变换就是对一个矩阵执行的以下三种变换操作之一:

- (1) 两行相互交换。
- (2) 一行和不为 0 的系数相乘。
- (3) 把一行与一个不为 0 的系数相乘后的和加到另一行上。

以方程(2.11)给出的矩阵为例,系数矩阵和常数向量并置后得到的增广矩阵为

$$\left[\begin{array}{ccc|c} 3 & 2 & -3 & 5 \\ 4 & -3 & 6 & 1 \\ 1 & 0 & -1 & 3 \end{array} \right] \quad (2.12)$$

对线性方程组的增广矩阵进行的行变换并不会影响线性方程组的解,只是使线性方程组更加容易计算。当用行变换求解一个线性方程组时,目标是把系数矩阵变换为它的最简形式,见定义 2.4。

【定义 2.4】 当且仅当一个矩阵满足以下条件时称之为最简形式:

- (1) 对每一个非 0 行,它的最左边的非 0 元素,即前导元素,必须为 1。
- (2) 每一个非 0 行均在全为 0 行的前面,也就是全 0 行在矩阵的底部。
- (3) 如果一行的前导元素在第 j 列,那么其他行在第 j 列的元素均为 0。

(4)对任意两个非0行 i_2 和 i_1 , 如果满足 $i_2 > i_1$, 并且相应的前导元素分别位于 j_2 和 j_1 列, 则必然存在 $j_2 > j_1$ 。

这个定义说明, 对于最简形式的矩阵, 其前导元素是向下向右移动的。进一步而言, 就是对于包含前导元素的列仅在前导元素的位置有一个为1的元素, 其他元素都为0。

【例 2.5】 以下矩阵即为最简形式的矩阵:

$$\begin{bmatrix} 1 & 0 & -3 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.13)$$

但是, 以下矩阵不是最简形式:

$$\begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.14)$$

这是因为第三行的前导元素不在第二行的前导元素的右边, 而且包含第四行前导元素的第四列中其他元素并不全为0。

算法 2.6 给出了把一个线性方程组的增广矩阵变换为最简形式所需的基本行变换。

【算法 2.6】 此算法把代表一个线性方程组的 $n \times (n+1)$ 增广矩阵 M 变换为最简形式。在以下的每一步, M 代表矩阵的当前状态, 而不是原始状态。

- (1) 令行下标 $i=1$;
- (2) 令列下标 $j=1$, 从第一列到第 n 列进行循环;
- (3) 找到第一个满足行号 $k \geq i$, 并且 $M_{kj} \neq 0$ 的行, 如果不存在, 则跳到步骤(8);
- (4) 如果 $k \neq i$, 那么就用定义 2.3 的行变换操作(1)交换 k 行和 i 行;
- (5) 将 i 行乘以 $1/M_{ij}$, 使矩阵 M 的元素 (i, j) 变为 1;
- (6) 利用基本行变换操作(3), 对于每一行 $r, 1 \leq r \leq n$ 且 $r \neq i$, 将第 i 行乘 $-M_{rj}$ 加到 r 行上。这一步将把第 j 列中除第 i 行之外的元素变为 0;
- (7) i 加 1;
- (8) 如果 $j < n$, j 加 1, 跳到步骤(3)。

下例说明了如何把算法 2.6 应用到式(2.11)所示的非齐次线性方程组求解过程中, 在增广系数矩阵变为最简后, 方程组的解就变得显而易见了。

【例 2.7】 求解以下非齐次线性方程组:

$$\begin{bmatrix} 3 & 2 & -3 \\ 4 & -3 & 6 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \\ 3 \end{bmatrix} \quad (2.15)$$

解 首先构造系数增广矩阵为

$$\left[\begin{array}{ccc|c} 3 & 2 & -3 & 5 \\ 4 & -3 & 6 & 1 \\ 1 & 0 & -1 & 3 \end{array} \right] \quad (2.16)$$

第一行数乘以 $1/3$, 产生前导元素 1

$$\xrightarrow{\text{第一行乘以 } 1/3} \left[\begin{array}{ccc|c} 1 & 2/3 & -1 & 5/3 \\ 4 & -3 & 6 & 1 \\ 1 & 0 & -1 & 3 \end{array} \right] \quad (2.17)$$

应用算法 2.6 的步骤(8), 消除第一列中的其他非 0 元素

$$\xrightarrow{\begin{array}{l} \text{第 1 行乘以 } -4 \text{ 加到第二行} \\ \text{第 1 行乘以 } -1 \text{ 加到第三行} \end{array}} \left[\begin{array}{ccc|c} 1 & 2/3 & -1 & 5/3 \\ 0 & -17/3 & 10 & -17/3 \\ 0 & -2/3 & 0 & 4/3 \end{array} \right] \quad (2.18)$$

处理第二行, 给该行乘以 $-\frac{3}{17}$ 得该行的前导元素 1

$$\xrightarrow{\text{第二行乘以 } -3/17} \left[\begin{array}{ccc|c} 1 & 2/3 & -1 & 5/3 \\ 0 & 1 & -30/17 & 1 \\ 0 & -2/3 & 0 & 4/3 \end{array} \right] \quad (2.19)$$

应用算法 2.6 的步骤(8), 消除第 2 列中的其他非 0 元素

$$\xrightarrow{\begin{array}{l} \text{第二行乘以 } -2/3 \text{ 加到第一行} \\ \text{第二行乘以 } 2/3 \text{ 加到第三行} \end{array}} \left[\begin{array}{ccc|c} 1 & 0 & 3/17 & 1 \\ 0 & 1 & -30/17 & 1 \\ 0 & 0 & -20/17 & 2 \end{array} \right] \quad (2.20)$$

最后, 对第三行采取相同的步骤

$$\xrightarrow{\text{第三行乘以 } -17/20} \left[\begin{array}{ccc|c} 1 & 0 & 3/17 & 1 \\ 0 & 1 & -30/17 & 1 \\ 0 & 0 & 1 & -17/10 \end{array} \right]$$

$$\xrightarrow{\text{第三行乘以 } -3/17 \text{ 加到第一行}} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 13/10 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & -17/10 \end{array} \right] \quad (2.21)$$

现在系数矩阵已经变成了最简形式。用简化的增广矩阵表示原来的方程组, 即

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & 13/10 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & -17/10 \end{array} \right] \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 13/10 \\ -2 \\ -17/10 \end{bmatrix} \quad (2.22)$$

从上式, 立即可以得出原方程组的解, 即

$$\begin{aligned} x &= 13/10 \\ y &= -2 \\ z &= -17/10 \end{aligned} \quad (2.23)$$

从上例中发现, 系数矩阵的最简形式就是单位阵, 在这种情况下, 相应的线性方程组有惟一解。如果最简系数矩阵有一行或多行为 0, 相应的线性方程组要么没有解要么有无穷多解。如果常数向量中与系数矩阵中全为 0 的一行对应的元素不为 0, 则方程组无解, 因为在该行将数值 0 等于了一个非 0 的数值。最后一种情况就是, 常数向量中的元素为 0, 则线性方程组相应的方程一定是用任意常数项表示的, 因此有无穷多个解。以任意常数项表示的方程的数量等于元素全为 0 的行的数量。可为任意常数的方程变量是与最简系数矩阵中不含前导元素的列对应的变量。

【例 2.8】 解以下齐次线性方程组

$$\begin{aligned} 2x + y + 3z &= 0 \\ y - z &= 0 \end{aligned}$$

$$x + 3y - z = 0 \quad (2.24)$$

解 齐次方程组的增广矩阵为

$$\left[\begin{array}{ccc|c} 2 & 1 & 3 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 3 & -1 & 0 \end{array} \right] \quad (2.25)$$

用算法 2.6 可以得出其最简形式

$$\left[\begin{array}{ccc|c} 1 & 0 & 2 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right] \quad (2.26)$$

因为此矩阵有一行全为 0, 可以给相应的变量分配一个任意值, 因为第三列矩阵元素中不含前导元素, 所以对应变量为第三个变量 z , 假设 $z = a$, 则前两行表示的方程式为

$$\begin{aligned} x + 2a &= 0 \\ y - a &= 0 \end{aligned} \quad (2.27)$$

因此, 此方程组的解为

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = a \begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix} \quad (2.28)$$

齐次线性方程组至少有一组解, 即 0 向量。只有系数矩阵的最简形式至少有一行全为 0 时才出现特殊解的情况。

2.3 逆矩阵

对一个 $n \times n$ 矩阵 M 而言, 如果存在一个矩阵 M^{-1} , 使 $MM^{-1} = M^{-1}M = I$, 则称矩阵 M 是可逆的。矩阵 M^{-1} 叫做 M 的逆矩阵。并不是每个矩阵都可逆, 没有逆矩阵的矩阵叫做奇异矩阵。例如, 任何有一行或一列为 0 的矩阵就是奇异矩阵。

【定理 2.9】 有一行或一列全为 0 的矩阵是不可逆的。

证明 假设 $n \times n$ 矩阵 F 的第 r 行元素全为 0, 对任一 $n \times n$ 矩阵 G , 两矩阵 FG 之乘积的第 (r, r) 元素可表示为 $\sum_{k=1}^n F_{rk}G_{kr}$ 。由于矩阵元素 F_{rk} 为 0, 则 FG 第 (r, r) 元素也为 0。另外, 因为矩阵 F 的逆阵与 F 的乘积矩阵的第 (r, r) 元素应该为 1, 所以 F 是不可逆的。同理, 有一列元素全为 0 的矩阵也不可逆。

利用这个定理, 在本节的后面会看到, 对于任意矩阵, 如果有一行是其他行的线性组合, 则这个矩阵是奇异的。如果有一列元素也满足该条件, 矩阵同样为奇异矩阵, 用以下定理可证明这一点。

【定理 2.10】 一个矩阵 M 是可逆的当且仅当 M^T 是可逆的。

证明 假定 M 是可逆的, 那么 M^{-1} 存在, 则有

$$M^T(M^{-1})^T = (M^{-1}M)^T = I^T = I \quad (2.29)$$

而且

$$(M^{-1})^T M^T = (MM^{-1})^T = I^T = I \quad (2.30)$$

因此, $(M^{-1})^T$ 是 M^T 的逆阵。类似, 如果假设 M^T 是可逆的, 那么 $(M^T)^{-1}$ 存在

$$M[(M^T)^{-1}]^T = [(M^T)^{-1}M^T]^T = I^T = I \quad (2.31)$$

而且

$$[(M^T)^{-1}]^T M = [M^T(M^T)^{-1}]^T = I^T = I \quad (2.32)$$

因此, $[(M^T)^{-1}]^T$ 是 M 的逆矩阵。

在讨论逆矩阵的计算方法之前, 先来熟悉以下几个定理。

【定理 2.11】 如果 F 和 G 是 $n \times n$ 的可逆矩阵, 那么 FG 也是可逆的且 $(FG)^{-1} = G^{-1}F^{-1}$ 。

证明 可以通过直接计算来证明这个定理, 利用矩阵乘法的组合性

$$G^{-1}F^{-1}(FG) = G^{-1}(F^{-1}F)G = G^{-1}G = I \quad (2.33)$$

一种类似于求矩阵最简形式的方法(参考算法 2.6)也可以用来计算矩阵的逆阵。为了计算 $n \times n$ 矩阵的逆阵, 首先在矩阵 M 的右边并接一个单位阵以形成一个 $n \times 2n$ 的矩阵 \tilde{M}

$$\tilde{M} = \left[\begin{array}{cccc|cccc} M_{11} & M_{12} & \cdots & M_{1n} & 1 & 0 & \cdots & 0 \\ M_{21} & M_{22} & \cdots & M_{2n} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ M_{n1} & M_{n2} & \cdots & M_{nn} & 0 & 0 & \cdots & 1 \end{array} \right] \quad (2.34)$$

对整个 \tilde{M} 执行矩阵基本行变换操作, 直到其左边的 $n \times n$ 矩阵变为单位阵, 这时右边 $n \times n$ 部分就是 M 的逆矩阵。计算过程见算法 2.12。

【算法 2.12】 这个算法用于计算 $n \times n$ 矩阵 M 的逆矩阵。其步骤如下:

- (1) 构造等式 2.34 中给出的增广矩阵 \tilde{M} , 在此算法中, \tilde{M} 指增广矩阵的当前状态, 而不是原始状态;
- (2) 令列下标 $j=1$, 从第一列到第 n 列进行循环;
- (3) 找到第一个满足行号 $i \geq j$, 并且 $\tilde{M}_{ij} \neq 0$ 的行, 如果不存在, 则 M 不可逆;
- (4) 如果 $i \neq j$, 那么就用定义 2.3 的行变换操作(1)交换 i 行和 j 行;
- (5) 将 j 行乘以 $1/\tilde{M}_{jj}$, 使矩阵 \tilde{M} 的 (j, j) 元素变为 1;
- (6) 用定义 2.3 基本行变换操作(3), 对于每一行 $r, 1 \leq r \leq n$ 且 $r \neq j$, 将第 j 行乘 $-\tilde{M}_{rj}$ 加到 r 行上, 这一步将把第 j 列中除第 j 行之外的元素变为 0;
- (7) 如果 $j < n$, j 加 1, 跳到步骤(3)。

算法 2.12 是计算矩阵的逆阵的直接算法, 而且可以确定一个矩阵是否可逆。下例说明了实现该算法的详细步骤。

【例 2.13】 计算 3×3 矩阵 M 的逆矩阵

$$M = \begin{bmatrix} 2 & 3 & 8 \\ 6 & 0 & -3 \\ -1 & 3 & 2 \end{bmatrix} \quad (2.35)$$

解 并接矩阵 M 和单位阵得

$$\tilde{M} = \left[\begin{array}{ccc|ccc} 2 & 3 & 8 & 1 & 0 & 0 \\ 6 & 0 & -3 & 0 & 1 & 0 \\ -1 & 3 & 2 & 0 & 0 & 1 \end{array} \right] \quad (2.36)$$

现在设 $j=1$, 按算法 2.12 的步骤(3)~(6)对矩阵 \tilde{M} 进行变换

$$\begin{array}{l}
 \xrightarrow{\text{第一行乘以 } 1/2} \\
 \xrightarrow{\substack{\text{第一行加到第三行} \\ \text{第一行乘以 } -6 \text{ 加到第二行}}}
 \end{array}
 \left[\begin{array}{ccc|ccc}
 1 & 3/2 & 4 & 1/2 & 0 & 0 \\
 6 & 0 & -3 & 0 & 1 & 0 \\
 -1 & 3 & 2 & 0 & 0 & 1 \\
 \hline
 1 & 3/2 & 4 & 1/2 & 0 & 0 \\
 0 & -9 & -27 & -3 & 1 & 0 \\
 0 & 9/2 & 6 & 1/2 & 0 & 1
 \end{array} \right] \quad (2.37)$$

令 $j=2$, 按算法 2.12 的步骤(3)~(6)对当前矩阵 \tilde{M} 进行变换

$$\begin{array}{l}
 \xrightarrow{\text{第二行乘以 } -1/9} \\
 \xrightarrow{\substack{\text{第二行乘以 } -3/2 \text{ 加到第一行} \\ \text{第二行以 } -9/2 \text{ 加到第三行}}}
 \end{array}
 \left[\begin{array}{ccc|ccc}
 1 & 3/2 & 4 & 1/2 & 0 & 0 \\
 0 & 1 & 3 & 1/3 & -1/9 & 0 \\
 0 & 9/2 & 6 & 1/2 & 0 & 1 \\
 \hline
 1 & 0 & -1/2 & 0 & 1/6 & 0 \\
 0 & 1 & 3 & 1/3 & -1/9 & 0 \\
 0 & 0 & -15/2 & -1 & 1/2 & 1
 \end{array} \right] \quad (2.38)$$

最后, 令 $j=3$, 重复上述步骤

$$\begin{array}{l}
 \xrightarrow{\text{第三行乘以 } -2/15} \\
 \xrightarrow{\substack{\text{第三行乘以 } 1/2 \text{ 加到第一行} \\ \text{第三行乘以 } -3 \text{ 加到第二行}}}
 \end{array}
 \left[\begin{array}{ccc|ccc}
 1 & 0 & -1/2 & 0 & 1/6 & 0 \\
 0 & 1 & 3 & 1/3 & -1/9 & 0 \\
 0 & 0 & 1 & 2/15 & -1/15 & -2/15 \\
 \hline
 1 & 0 & 0 & 1/15 & 2/15 & -1/15 \\
 0 & 1 & 0 & -1/15 & 4/45 & 2/5 \\
 0 & 0 & 1 & 2/15 & -1/15 & -2/15
 \end{array} \right] \quad (2.39)$$

现在, 右侧的 3×3 矩阵就是 M 的逆矩阵, 即

$$M^{-1} = \frac{1}{45} \begin{bmatrix} 3 & 6 & -3 \\ -3 & 4 & 18 \\ 6 & -3 & 6 \end{bmatrix} \quad (2.40)$$

定理 2.14 说明了为什么算法 2.12 可以求出一个矩阵的逆矩阵。

【定理 2.14】 假设 $n \times n$ 矩阵 M' 是 $n \times n$ 矩阵 M 进行基本行变换后得到的矩阵, 则有 $M' = EM$, 其中 E 是对单位阵进行相同的基本行变换的最终矩阵。

证明 以下分别给出定义 2.3 中 3 个基本行变换的证明。

(1) 假设矩阵 E 是单位阵的 r 行和 s 行交换后的矩阵, 则 E 的元素为

$$E_{ij} = \begin{cases} \delta_{ij}, & i \neq r \text{ 且 } i \neq s \\ \delta_{sj}, & i = r \\ \delta_{rj}, & i = s \end{cases} \quad (2.41)$$

其中 δ_{ij} 是等式(1.42)定义的克罗内克符号, 则积 EM 的元素为

$$(EM)_{ij} = \sum_{k=1}^n E_{ik} M_{kj} = \begin{cases} M_{ij}, & i \neq r \text{ 且 } i \neq s \\ M_{sj}, & i = r \\ M_{rj}, & i = s \end{cases} \quad (2.42)$$

可知矩阵 M 的 r 行和 s 行已经交换位置。

(2) 假设 E 是单位阵的 r 行乘以比例系数 a 后的结果, 则 E 的元素为

$$E_{rj} = \begin{cases} \delta_{ij}, & i \neq r \\ a\delta_{ij}, & i = r \end{cases} \quad (2.43)$$

EM 的元素可由下式相应给出

$$(EM)_{ij} = \sum_{k=1}^n E_{ik}M_{kj} = \begin{cases} M_{ij}, & i \neq r \\ aM_{ij}, & i = r \end{cases} \quad (2.44)$$

因此,矩阵 M 的 r 行也乘了比例系数 a 。

(3) 假设 E 是单位阵的 r 行乘以系数 a 加到 s 行后的结果。那么 E 的元素可表示为

$$E_{ij} = \begin{cases} \delta_{ij}, & i \neq s \\ \delta_{ij} + a\delta_{rj}, & i = s \end{cases} \quad (2.45)$$

EM 的元素可表示为

$$(EM)_{ij} = \sum_{k=1}^n E_{ik}M_{kj} = \begin{cases} M_{ij}, & i \neq s \\ M_{ij} + aM_{rj}, & i = s \end{cases} \quad (2.46)$$

因此,矩阵 M 的 r 行乘以 a 并加到了 s 行上。

把单位阵经过行变换后得到的矩阵称为初等矩阵。如果为了把矩阵 M 变为单位阵而必须进行 k 次基本行变换,那么

$$I = E_k E_{k-1} \cdots E_1 M \quad (2.47)$$

其中矩阵 E_1, E_2, \dots, E_k 是单位阵经过 k 次行变换后所得的相对应的初等矩阵。也就是说矩阵 $E_k E_{k-1} \cdots E_1$ 之间的积即为 M 的逆矩阵,这其实就是对等式(2.34)中与矩阵 M 对应的单位阵进行 k 次行变换所得到的结果。

如果矩阵 M 是奇异的,那么不可能找到初等矩阵 E_1, E_2, \dots, E_k 满足等式(2.47)。这是因为奇异矩阵的行形成了一个线性相关集合,如定理 2.15 所示。

【定理 2.15】 当且仅当矩阵 M 的行是一个线性无关向量集时, $n \times n$ 矩阵 M 是可逆的。

证明 把矩阵 M 的行记作 $R_1^T, R_2^T, \dots, R_n^T$, 分两部分证明此定理。

(1) 为了证明如果矩阵 M 是可逆的,那么它的行就是一个线性无关向量集,证明该命题的等价逆命题,也就是如果矩阵 M 的行向量是线性相关集,那么矩阵 M 必然是奇异的。因此,假定矩阵 M 的行是线性相关的,那么必然存在行 r 是其他 k 行的线性组合,表示为

$$R_r^T = a_1 R_{s_1}^T + a_2 R_{s_2}^T + \cdots + a_k R_{s_k}^T \quad (2.48)$$

其中 a_i 是比例系数, s_i 是矩阵 M 中除 r 行以外其他 k 行的索引。假设 $n \times n$ 矩阵 E_i 是行 s_i 乘以 a_i 加到行 r 后的初等矩阵,则有

$$M = E_k E_{k-1} \cdots E_1 M' \quad (2.49)$$

其中, M' 是矩阵 M 的 r 行所有元素用 0 取代后所得到的矩阵,由定理 2.9 得,矩阵 M' 是奇异的,因此矩阵 M 也是奇异的。

(2) 现在假定矩阵 M 的行向量集是线性无关的,又知在一个矩阵上执行基本行变换并不会改变行之间的线性无关性。执行算法 2.12,如果步骤(3)失败了,又由于从 R_j^T 行到 R_n^T 行至少存在一个非 0 元素的列数小于行数,则矩阵的 j 行到 n 行形成了一个线性相关集,这与前面的假定产生矛盾,所以步骤(3)不可能失败,因此 M 是可逆的。

逆的。

这个定理告诉我们,每一个奇异矩阵都能够写成一个初等矩阵和一行元素都为0的矩阵的组合。结合2.4节中介绍的行列式,利用该性质可以进行奇异性测试。

2.4 行列式

方阵的行列式是从方阵的元素得到的一个标量,矩阵 \mathbf{M} 的行列式记作 $\det \mathbf{M}$ 。在表示矩阵的元素时,用垂直线取代矩阵两侧的括号,以表明计算的对象是该矩阵的行列式。例如 3×3 矩阵 \mathbf{M} 的行列式为

$$\det \mathbf{M} = \begin{vmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{vmatrix} \quad (2.50)$$

$n \times n$ 矩阵 \mathbf{M} 的行列式的值由一个递推公式给出。为了便于表示,用符号 $\mathbf{M}^{[i,j]}$ 表示将原始矩阵 \mathbf{M} 中的第 i 行和第 j 列删除后形成的 $(n-1) \times (n-1)$ 矩阵。例如,假设 \mathbf{M} 是以下的 3×3 矩阵

$$\mathbf{M} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad (2.51)$$

那么, $\mathbf{M}^{[2,3]}$ 就是以下的 2×2 矩阵

$$\mathbf{M}^{[2,3]} = \begin{bmatrix} 1 & 2 \\ 7 & 8 \end{bmatrix} \quad (2.52)$$

用这种方法, $n \times n$ 矩阵的行列式值的计算方法可表示如下。首先,定义 1×1 矩阵的行列式就是矩阵的元素自己。然后, $n \times n$ 矩阵的行列式可由以下两个公式给出,即

$$\det \mathbf{M} = \sum_{i=1}^n (-1)^{i+k} M_{ik} \det \mathbf{M}^{[i,k]} \quad (2.53)$$

$$\det \mathbf{M} = \sum_{j=1}^n (-1)^{k+j} M_{kj} \det \mathbf{M}^{[k,j]} \quad (2.54)$$

其中 k 是满足条件 $1 \leq k \leq n$ 的任意常数。很明显,上述两式都给出了与 k 无关的相同的行列式值。

由公式(2.53)和公式(2.54)可以很容易地得出二阶矩阵的行列式公式:

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc \quad (2.55)$$

同时,利用这两个公式也能直接给出三阶矩阵的行列式的计算公式,令式(2.54)中 $k=1$ 可得

$$\begin{aligned} \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} &= a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \\ &= a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + \end{aligned}$$

$$a_{13}(a_{21}a_{32} - a_{22}a_{31}) \quad (2.56)$$

很明显,单位阵 I_n 的行列式是 1,这个结果用公式(2.54)并令 $k=1$ 可直接计算出来。

通过研究基本行变换对矩阵的行列式的影响,可以得到一些有用的信息,利用这些信息计算矩阵的行列式比直接使用公式(2.53)和公式(2.54)要简便得多。

【定理 2.16】 执行基本行变换对一个矩阵的行列式有以下性质:

- (1)两行相互交换则行列式取反;
- (2)矩阵中的一行乘以比例系数 a 则行列式也变为原来的 a 倍;
- (3)把一行的若干倍加到另一行对行列式没有影响。

证明

- (1)用归纳法来证明这个定理。由于行交换不适用于一阶矩阵,所以从二阶矩阵开始。利用公式(2.55)可直接计算出二阶矩阵的行列式

$$\begin{vmatrix} c & d \\ a & b \end{vmatrix} = cb - ad = -(ad - cb) = \begin{vmatrix} a & b \\ c & d \end{vmatrix} \quad (2.57)$$

可见对二阶行列式来说,该定理成立。对于 $n \times n$ 矩阵,假定对直到 $(n-1) \times (n-1)$ 的矩阵本定理都成立。假定矩阵 F 的 r 行和 s 行交换后得矩阵 G 。选定另一行 k ($k \neq r$ 并且 $k \neq s$),由式(2.54)得出

$$\det G = \sum_{j=1}^n (-1)^{k+j} G_{kj} \det G^{[k,j]} \quad (2.58)$$

因为 $G^{[j,k]}$ 是一个 $(n-1) \times (n-1)$ 的矩阵,通过归纳可得出对于每一个 j 均有 $\det G^{[j,k]} = -\det F^{[j,k]}$,因此, $\det G = -\det F$ 。

- (2)假定矩阵 G 表示矩阵 F 的行 k 乘以比例系数 a 后的结果,由式(2.54)得

$$\begin{aligned} \det G &= \sum_{j=1}^n (-1)^{k+j} G_{kj} \det G^{[k,j]} \\ &= \sum_{j=1}^n (-1)^{k+j} a F_{kj} \det F^{[k,j]} \end{aligned} \quad (2.59)$$

因此, $G = a \det F$ 。

在证明性质(3)之前,需要性质(1)的以下推论。

【推论 2.17】 如果矩阵有相同的两行,则其行列式为 0。

证明 假定矩阵 M 有相同的两行,如果把这两行交换,矩阵并没有发生任何变化,但是行列式的符号已经变了,也就是 $\det M = -\det M$,因此 $\det M = 0$ 。

证明定理 2.16(3) 令 G 表示矩阵 F 的 r 行乘以 a 倍加到 k 行后所得的矩阵,那么由式(2.54)可得出

$$\begin{aligned} \det G &= \sum_{j=1}^n (-1)^{k+j} G_{kj} \det G^{[k,j]} = \sum_{j=1}^n (-1)^{k+j} (F_{kj} + aF_{rj}) \det F^{[k,j]} \\ &= \det F + a \sum_{j=1}^n (-1)^{k+j} F_{rj} \det F^{[k,j]} \end{aligned} \quad (2.60)$$

$\sum_{j=1}^n (-1)^{k+j} F_{rj} \det F^{[k,j]}$ 是矩阵 F 的 k 行元素被 r 行元素取代后所得矩阵的行列式。因此此矩阵有两个相同的行,由推论 2.17 知 $\det G = \det F$ 。

因为初等矩阵其实就是在单位阵上执行基本行变换所得,所以从定理 2.16 可以推出它们的行列式。初等矩阵经过一次行交换后的行列式是 -1 。初等矩阵中的一行乘以比例系数 a 后,其行列式为 a 。初等矩阵中一行的倍数加到另一行上所得矩阵的行列式是 1 。这些都是矩阵执行相应的基本行变换时其行列式应乘的数值。因此能推断出,如果 E 是 $n \times n$ 初等矩阵,那么 $\det EM = \det E \det M$,其中 M 是任意 $n \times n$ 矩阵,也就是矩阵 E 与矩阵 M 相乘就相当于矩阵 M 执行基本行变换。利用这个结果可以得出以下两个重要定理。

【定理 2.18】 $n \times n$ 矩阵 M 当且仅当 $\det M \neq 0$ 时是不可逆的。

证明 假定矩阵 M 不可逆,那么 M 就可看作一系列初等矩阵的乘积,其中每一个矩阵的行列式均不为 0 。因为一系列初等矩阵乘积的行列式等于这些矩阵的行列式的乘积,所以矩阵 M 的行列式不为 0 。现在假定 M 是奇异矩阵,则矩阵 M 的行必然是线性相关的,那么 M 可写作一系列初等矩阵和一个有一行全为 0 的矩阵的乘积。因为矩阵有一行为 0 ,所以其行列式必然为 0 ,则该矩阵与其他矩阵的乘积的行列式也是 0 。

【定理 2.19】 对任意两个 $n \times n$ 的矩阵 F 和 G , $\det FG = \det F \det G$ 。

证明 假设 F 和 G 中有一个是奇异矩阵,则 FG 也是奇异的,那么,因为方程两边均为 0 ,所以等式成立。否则,矩阵 F 和 G 就可以被完全分解为初等矩阵,因为初等矩阵乘积的行列式等于行列式的乘积,所以等式成立。

定理 2.18 给出一个检验矩阵奇异性的方法。如果已知 $n \times n$ 矩阵 M 的行列式不为 0 ,用以下公式可以计算逆矩阵 M^{-1} 的每个元素。

【定理 2.20】 假设 F 是 $n \times n$ 矩阵,如果另一 $n \times n$ 矩阵 G 的元素为

$$G_{ij} = (-1)^{i+j} \frac{\det F^{[j, i]}}{\det F} \quad (2.61)$$

那么 $G = F^{-1}$ 。

证明 根据 FG 的乘积公式,可得

$$\begin{aligned} (FG)_{ij} &= \sum_{k=1}^n F_{ik} G_{kj} = \sum_{k=1}^n F_{ik} (-1)^{k+j} \frac{\det F^{[j, k]}}{\det F} \\ &= \frac{1}{\det F} \sum_{k=1}^n (-1)^{k+j} F_{ik} \det F^{[j, k]} \end{aligned} \quad (2.62)$$

如果 $i=j$,那么式(2.62)中矩阵 F 的行列式等于式(2.53),故式(2.62)再乘以 $1/\det F$ 就得到 $(FG)_{ij} = 1$ 。如果 $i \neq j$,上式行列式就为矩阵 F 中 j 行被 i 行替代后的矩阵的行列式。因为矩阵中有两行相同,其行列式必然为 0 ,因此 $(FG)_{ij} = 0$ 。因为矩阵 FG 的主对角线元素是 1 而其余的元素为 0 ,所以 FG 是单位阵。同理可以证明 GF 也是单位阵,所以 $G = F^{-1}$ 成立。

从等式(2.61),可以得出计算机图形学中经常用到的几个逆矩阵的计算公式,如式(2.63)和式(2.64)。 2×2 矩阵 A 的逆矩阵为

$$A^{-1} = \frac{1}{\det A} \begin{bmatrix} A_{22} & -A_{12} \\ -A_{21} & A_{11} \end{bmatrix} \quad (2.63)$$

3×3 矩阵 B 的逆矩阵为

$$B^{-1} = \frac{1}{\det B} \begin{bmatrix} B_{22}B_{33} - B_{23}B_{32} & B_{13}B_{32} - B_{12}B_{33} & B_{12}B_{23} - B_{13}B_{22} \\ B_{23}B_{31} - B_{21}B_{33} & B_{11}B_{33} - B_{13}B_{31} & B_{13}B_{21} - B_{11}B_{23} \\ B_{21}B_{32} - B_{22}B_{31} & B_{12}B_{31} - B_{11}B_{32} & B_{11}B_{22} - B_{12}B_{21} \end{bmatrix} \quad (2.64)$$

2.5 特征值与特征向量

对于可逆方阵来说,必然存在一个向量,当该向量与可逆方阵相乘时,向量只会发生大小的变化,而方向不变。也就是说,对于 $n \times n$ 矩阵 M ,存在着非 0 的 n 维向量 V_1, V_2, \dots, V_n 满足

$$MV_i = \lambda_i V_i \quad (2.65)$$

其中,比例系数 λ_i 称为矩阵 M 的特征值,而向量 V_i 则是相应于这些特征值的特征向量。

通过对式(2.65)的变换可得到矩阵的特征值

$$(M - \lambda_i I)V_i = 0 \quad (2.66)$$

其中 I 是 $n \times n$ 的单位阵。对于非 0 向量 V_i ,如果上式成立,矩阵 $M - \lambda_i I$ 一定是奇异矩阵。否则,可以对 $M - \lambda_i I$ 求逆

$$V_i = (M - \lambda_i I)^{-1} 0 = 0 \quad (2.67)$$

这和假设 $V_i \neq 0$ 矛盾。因为 $M - \lambda_i I$ 是奇异的,它的行列式必定为 0,因此通过求解以下方程可以计算出特征值 λ_i

$$\det(M - \lambda I) = 0 \quad (2.68)$$

由式 2.68 得出的关于 λ 的 n 阶多项式叫做矩阵 M 的特征多项式。这个多项式的根就是矩阵 M 的特征值。

【例 2.21】 计算以下矩阵的特征值

$$M = \begin{bmatrix} 1 & 1 \\ 3 & -1 \end{bmatrix} \quad (2.69)$$

解 矩阵 $M - \lambda_i I$ 为

$$M - \lambda I = \begin{bmatrix} 1 - \lambda & 1 \\ 3 & -1 - \lambda \end{bmatrix}$$

计算 $M - \lambda I$ 的行列式,得矩阵 M 特征多项式为

$$(1 - \lambda)(-1 - \lambda) - 3 \quad (2.70)$$

化简这个多项式并令其等于 0,则有

$$\lambda^2 - 4 = 0 \quad (2.71)$$

解得, $\lambda_1 = 2$ 和 $\lambda_2 = -2$,即为矩阵 M 的两个特征值。

一旦矩阵的特征值求得,通过求解等式(2.66)给出的齐次方程组就可以计算出相应的特征向量。因为矩阵 $M - \lambda_i I$ 是奇异的,它的简化形式至少有一行全为 0,所以有无穷多解。等式(2.65)的一个明显的特性就是如果 V_i 是对应于特征值的特征向量,那么 V_i 与任何比例系数的乘积 αV_i 也是一个特征向量。因此,特征向量可以表示成任意常数的形式,如果有必要也可以取特征向量的长度为 1。

【例 2.22】 计算以下矩阵的特征向量

$$M = \begin{bmatrix} 1 & 1 \\ 3 & -1 \end{bmatrix} \quad (2.72)$$

解 在例 2.21 中已经计算出矩阵 M 的特征值分别是 2 和 -2,通过求解相应的齐次方

程组 $(M - \lambda_1 I)V_1 = 0$, 可得到该矩阵的特征向量。对于特征值 $\lambda_1 = 2$ 有

$$\begin{bmatrix} -1 & 1 \\ 3 & -3 \end{bmatrix} V_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2.73)$$

对于特征值 $\lambda_2 = -2$, 则有

$$\begin{bmatrix} 3 & 1 \\ 3 & 1 \end{bmatrix} V_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2.74)$$

上述方程组的解分别为

$$V_1 = a \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad V_2 = b \begin{bmatrix} 1 \\ -3 \end{bmatrix} \quad (2.75)$$

其中 a, b 是任意不为 0 的常数。

通常, 如果由矩阵的特征多项式的根得出的特征值是复数, 则相应的特征向量也含有复数元素。确保有实数特征值以及相应的实数特征向量的矩阵是对称矩阵。

【定义 2.23】 一个 $n \times n$ 矩阵当且仅当对任意的 i 和 j 均有 $M_{ij} = M_{ji}$ 时是对称矩阵, 也就是, 如果一个矩阵的元素关于主对角线对称, 则该矩阵就称为对称矩阵。

对称矩阵的特征值和特征向量有以下两个性质。

【定理 2.24】 矩阵元素为实数的对称矩阵的特征值也是实数。

证明 假设 λ 和 V 是矩阵 M 的特征值和相应的特征向量, 且满足 $MV = \lambda V$ 。等式两边同时左乘行向量 \bar{V}^T 可得

$$\bar{V}^T M V = \bar{V}^T \lambda V = \lambda \bar{V}^T V \quad (2.76)$$

其中, 上划线表示复数的共轭, 即向量和矩阵中的元素的共轭。因为复数 $a + bi$ 和它的共轭复数 $a - bi$ 的乘积是实数 $a^2 + b^2$, 那么 $\bar{V}^T V$ 也是一个实数。如果能够证明 $\bar{V}^T M V$ 是一个实数, 那么就可以推断出 λ 也是实数。 $\bar{V}^T M V$ 的共轭为

$$\overline{\bar{V}^T M V} = V^T M \bar{V} \quad (2.77)$$

其中, 由于矩阵 M 是实矩阵, 所以 $\bar{M} = M$ 。同时, $\bar{V}^T M V$ 是一个 1×1 的矩阵, 它的转置矩阵就是它自己, 所以有

$$V^T M \bar{V} = (V^T M \bar{V})^T = \bar{V}^T M^T V \quad (2.78)$$

因为矩阵 M 是对称的, $M^T = M$, 所以有

$$\bar{V}^T M \bar{V} = \bar{V}^T M V \quad (2.79)$$

式(2.79)表明 $\bar{V}^T M V$ 等于它本身的共轭, 因而是一个实数。这就证明特征值 λ 必为实数。

【定理 2.25】 对应于对称矩阵 M 的不同特征值的特征向量是正交的。

证明 假设 λ_1 和 λ_2 是矩阵 M 的不同特征值, V_1 和 V_2 是相应的特征向量, 则有等式 $MV_1 = \lambda_1 V_1$ 和 $MV_2 = \lambda_2 V_2$ 。通过以下步骤, 可以证明 $\lambda_1 V_1^T V_2 = \lambda_2 V_1^T V_2$ 。

$$\begin{aligned} \lambda_1 V_1^T V_2 &= (\lambda_1 V_1)^T V_2 = (M V_1)^T V_2 \\ &= V_1^T M V_2 = \lambda_2 V_1^T V_2 \end{aligned} \quad (2.80)$$

其中用到了 $M^T = M$ 这一特性。于是可知

$$(\lambda_1 - \lambda_2) V_1^T V_2 = 0 \quad (2.81)$$

但是, 其中 λ_1, λ_2 是不同的, 因此必然有 $V_1^T V_2 = 0$ 。这恰好是 V_1 和 V_2 的点积 $V_1 \cdot V_2$, 所以特征向量是正交的。

2.6 对角化

对角矩阵就是只在主对角线上有非 0 元素的矩阵,也就是说,如果一个 $n \times n$ 矩阵 M 是对角阵,必然满足 $M_{ij} = 0$,其中 $i \neq j$ 。给定方阵 M ,如果能找到一个矩阵 A 使得 $A^{-1}MA$ 是对角矩阵,那么就说矩阵 A 将矩阵 M 对角化了。虽然一般情况下上述情况不一定成立,但是下面的定理表明,如果能找到 $n \times n$ 矩阵的 n 个线性无关特征向量,则该矩阵可对角化。

【定理 2.26】 假设 M 是特征值为 $\lambda_1, \lambda_2, \dots, \lambda_n$ 的 $n \times n$ 矩阵,且与特征值对应的特征向量 V_1, V_2, \dots, V_n 是一个线性无关集,那么由下式所示的矩阵 A

$$A = [V_1 \ V_2 \ \cdots \ V_n] \quad (2.82)$$

(其中,矩阵 A 的列是特征向量 V_1, V_2, \dots, V_n) 可以将矩阵 M 对角化,而且 $A^{-1}MA$ 的主对角线元素是矩阵 M 的特征值,即

$$A^{-1}MA = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \quad (2.83)$$

相反,如果存在一个可逆矩阵 A 使得 $A^{-1}MA$ 是一个对角阵,那么矩阵 A 的列必然是矩阵 M 的特征向量,而 $A^{-1}MA$ 的主对角线元素则是矩阵 M 的特征值。

证明 首先分析矩阵的乘积 MA 。既然 A 的 j 列是特征向量 V_j ,那么 MA 的 j 列就等于 MV_j 。由于 V_j 是特征向量,则 $MV_j = \lambda_j V_j$,因此 MA 可表示为

$$\begin{aligned} MA &= [\lambda_1 V_1 \ \lambda_2 V_2 \ \cdots \ \lambda_n V_n] \\ &= [V_1 \ V_2 \ \cdots \ V_n] \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \\ &= A \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \end{aligned} \quad (2.84)$$

因为特征向量 V_j 是线性无关的,且矩阵 A 可逆,则 $A^{-1}MA$ 可以表示为

$$A^{-1}MA = A^{-1}A \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}$$

$$= \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \quad (2.85)$$

现在证明逆定理,也就是可以将矩阵 M 对角化的可逆矩阵 A 一定是由矩阵 M 的特征向量组成。假定 D 是 $n \times n$ 的对角化矩阵,且对于 $n \times n$ 矩阵 A ,满足 $D = A^{-1}MA$,那么以下表达式成立

$$AD = MA \quad (2.86)$$

令 V_j 表示矩阵 A 的第 j 列, d_1, d_2, \dots, d_n 表示矩阵 D 的主对角线元素,那么 AD 可写为

$$\begin{aligned} AD &= [V_1 \ V_2 \ \cdots \ V_n] \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{bmatrix} \\ &= [d_1 V_1 \ d_2 V_2 \ \cdots \ d_n V_n] \end{aligned} \quad (2.87)$$

乘积 MA 为

$$MA = [MV_1 \ MV_2 \ \cdots \ MV_n] \quad (2.88)$$

其中,矩阵积 AD 的第 j 列和 MA 的 j 列相等,这表明 $MV_j = d_j V_j$,因此,每一个 V_j 是与矩阵 M 的特征值 d_j 对应的特征向量。

因为对称矩阵 M 的特征向量是正交的,列由 M 的单位特征向量组成的矩阵 A 是一个正交矩阵,且满足 $A^{-1} = A^T$ 。以对称矩阵 M 的特征值为对角元素组成的对角矩阵 D 可表示为

$$D = A^T M A \quad (2.89)$$

【例 2.27】 求矩阵 M 的对角化矩阵

$$M = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (2.90)$$

解 矩阵 M 的特征多项式是

$$\begin{aligned} \det(M - \lambda I) &= -\lambda^3 + 2\lambda^2 + 2\lambda - 1 \\ &= -(\lambda + 1)(\lambda^2 - 3\lambda + 1) \end{aligned} \quad (2.91)$$

以上特征多项式的根就是矩阵的特征值

$$\begin{aligned} \lambda_1 &= -1 \\ \lambda_2 &= \frac{3 + \sqrt{5}}{2} \\ \lambda_3 &= \frac{3 - \sqrt{5}}{2} \end{aligned} \quad (2.92)$$

与特征值 λ_1 对应的特征向量 V_1 可以通过求解以下的齐次线性方程组得到

$$\begin{bmatrix} 3 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} V_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.93)$$

变换系数矩阵可得

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{V}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.94)$$

相应的解为

$$\mathbf{V}_1 = a \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.95)$$

对于特征值 λ_2 , 需要求解以下方程组:

$$\begin{bmatrix} \frac{1-\sqrt{5}}{2} & 1 & 0 \\ 1 & \frac{-1-\sqrt{5}}{2} & 0 \\ 0 & 0 & \frac{-5-\sqrt{5}}{2} \end{bmatrix} \mathbf{V}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.96)$$

化简后可得

$$\begin{bmatrix} 1 & \frac{-1-\sqrt{5}}{2} & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{V}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.97)$$

则第二个特征向量可表示为

$$\mathbf{V}_2 = b \begin{bmatrix} \frac{1+\sqrt{5}}{2} \\ 1 \\ 0 \end{bmatrix} \quad (2.98)$$

同理, 特征向量 \mathbf{V}_3 等于

$$\mathbf{V}_3 = c \begin{bmatrix} \frac{1-\sqrt{5}}{2} \\ 1 \\ 0 \end{bmatrix} \quad (2.99)$$

选择适当的常数 a, b 和 c 使特征向量的大小为 1。由于矩阵 M 为对称矩阵, 所以这 3 个特征向量是正交的, 令矩阵 A 为

$$\mathbf{A} = \begin{bmatrix} \frac{\mathbf{V}_1}{\|\mathbf{V}_1\|} & \frac{\mathbf{V}_2}{\|\mathbf{V}_2\|} & \frac{\mathbf{V}_3}{\|\mathbf{V}_3\|} \end{bmatrix} \approx \begin{bmatrix} 0 & 0.851 & -0.526 \\ 0 & 0.526 & 0.851 \\ 1 & 0 & 0 \end{bmatrix} \quad (2.100)$$

矩阵 A 就是可以将矩阵 M 对角化的正交矩阵

$$\mathbf{A}^{-1} \mathbf{M} \mathbf{A} = \mathbf{A}^T \mathbf{M} \mathbf{A} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & \frac{3+\sqrt{5}}{2} & 0 \\ 0 & 0 & \frac{3-\sqrt{5}}{2} \end{bmatrix} \quad (2.101)$$

2.7 本章小结

矩阵乘积

如果 F 是一个 $n \times m$ 矩阵, G 是一个 $m \times p$ 矩阵, 则 FG 是一个 $n \times p$ 的矩阵, 其 (i, j) 元素可由下式求得

$$(FG)_{ij} = \sum_{k=1}^m F_{ik}G_{kj}$$

行列式

$n \times n$ 矩阵 M 的行列式可由以下公式求得

$$\det M = \sum_{i=1}^n (-1)^{i+k} M_{ik} \det M^{(i,k)}$$

$$\det M = \sum_{j=1}^n (-1)^{k+j} M_{kj} \det M^{(k,j)}$$

二阶矩阵的行列式为

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc$$

三阶矩阵的行列式为

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31})$$

逆矩阵

当且仅当 $n \times n$ 矩阵 M 的行是一个线性无关集时, 矩阵 M 是可逆的。也就是说, 当且仅当 $\det M \neq 0$, 矩阵 M 可逆。

$n \times n$ 矩阵 F 的逆阵 G 的元素可以用以下公式求出

$$G_{ij} = (-1)^{i+j} \frac{\det F^{(j,i)}}{\det F}$$

应用这个公式, 2×2 矩阵 A 的逆阵为

$$A^{-1} = \frac{1}{\det A} = \begin{bmatrix} A_{22} & -A_{12} \\ -A_{21} & A_{11} \end{bmatrix}$$

3×3 矩阵 B 的逆阵为

$$B^{-1} = \frac{1}{\det B} \begin{bmatrix} B_{22}B_{33} - B_{23}B_{32} & B_{13}B_{32} - B_{12}B_{33} & B_{12}B_{23} - B_{13}B_{22} \\ B_{23}B_{31} - B_{21}B_{33} & B_{11}B_{33} - B_{13}B_{31} & B_{13}B_{21} - B_{11}B_{23} \\ B_{21}B_{32} - B_{22}B_{31} & B_{12}B_{31} - B_{11}B_{32} & B_{11}B_{22} - B_{12}B_{21} \end{bmatrix}$$

特征值和特征向量

$n \times n$ 矩阵 M 的特征值为下式给出的矩阵特征多项式的根

$$\det(\mathbf{M} - \lambda \mathbf{I})$$

与矩阵 \mathbf{M} 的特征值对应的特征向量可通过计算以下齐次线性方程组的解而得到

$$(\mathbf{M} - \lambda \mathbf{I}) = \mathbf{0}$$

实数对称矩阵的特征值是实数。与实对称矩阵的不同特征值对应的特征向量是正交的。

对角化

如果 $\mathbf{V}_1, \mathbf{V}_2, \dots, \mathbf{V}_n$ 是 $n \times n$ 矩阵 \mathbf{M} 的线性无关的特征向量, 则由下式给出的矩阵 \mathbf{A} ,

$$\mathbf{A} = [\mathbf{V}_1 \quad \mathbf{V}_2 \quad \dots \quad \mathbf{V}_n]$$

可以将矩阵 \mathbf{M} 对角化, 即

$$\mathbf{A}^{-1} \mathbf{M} \mathbf{A} = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}$$

其中 $\lambda_1, \lambda_2, \dots, \lambda_n$ 是矩阵 \mathbf{M} 的特征值。

习 题

1. 计算以下矩阵的行列式

$$(1) \begin{bmatrix} 2 & 7 \\ -3 & 1/2 \end{bmatrix}$$

$$(2) \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$(3) \begin{bmatrix} 1/2 & \sqrt{3}/2 & 0 \\ -\sqrt{3}/2 & 1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$(4) \begin{bmatrix} 5 & 7 & 1 \\ 17 & 2 & 64 \\ 10 & 14 & 2 \end{bmatrix}$$

2. 计算以下矩阵的逆阵

$$(1) \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

$$(2) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 2 \\ 3 & 0 & 8 \end{bmatrix}$$

$$(3) \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$(4) \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. 解以下齐次线性方程组

$$\begin{cases} 4x + 3y + 2z = 0 \\ x - y - 3z = 0 \\ 2x + 3y + 4z = 0 \end{cases}$$

4. 计算以下矩阵的特征值

$$\begin{bmatrix} 2 & 0 & 0 \\ 5 & 2 & 3 \\ -4 & 3 & 2 \end{bmatrix}$$

5. 设 $n \times n$ 矩阵 M 的行由向量 $R_1^T, R_2^T, \dots, R_n^T$ 组成, 证明如果 M 的行是一个线性无关集, 那么矩阵 EM 的行也是一个线性无关集(其中 E 是初等矩阵)。
6. 上三角矩阵 M 是元素满足 $M_{ij} = 0$ (其中 $i > j$), 也就是主对角线以下的元素全为 0 的矩阵。证明上三角矩阵的行列式就等于其主对角线元素的乘积。
7. 令 D 是主对角线元素为 d_1, d_2, \dots, d_n 的 $n \times n$ 矩阵, 如下所示:

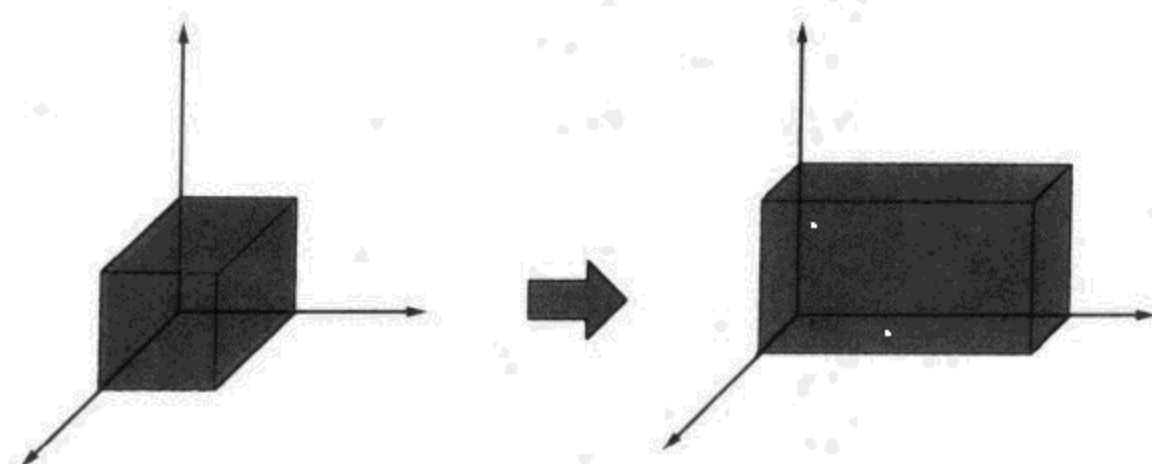
$$D = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{bmatrix}$$

证明 D 的逆阵仍然是对角矩阵且主对角线元素为 $1/d_1, 1/d_2, \dots, 1/d_n$ 。

8. 实现一个 3×3 矩阵的 C++ 封装类, 该类应该能存储矩阵的 9 个元素。除了执行初始化的默认构造函数外, 这个类应该有一个以 9 个浮点数为参数的构造函数, 并且用这些值初始化矩阵元素。这个类也应该有与标量之间的加、减、乘和除的重载运算符、与另一个 3×3 矩阵的乘法重载运算符以及与一个三维向量对象之间的乘法重载运算符(见第 1 章习题 4), 提供一个把矩阵初始化为单位矩阵式的函数。最后, 提供计算 3×3 矩阵行列式和逆矩阵的函数。



第3章 变换



在 3D 游戏的整个开发过程中,通常需要以某种方式对一系列的向量进行变换。通常用到的变换包括平移、缩放和旋转。对一组点的平移,只需将每个点简单地加上一个偏移向量就可以实现。统一缩放(即等量地缩放向量的每个坐标)可以通过数量乘积来实现。非统一缩放、旋转以及其他更复杂的变换则需要利用矩阵乘积来实现。

3.1 通用变换

通常可将 $n \times n$ 可逆矩阵 M 看成是从一个坐标系到另一个坐标系的变换矩阵。 M 的列给出了坐标轴从原坐标系到新坐标系的映射。例如,假设 M 是一个 $n \times n$ 可逆矩阵,当 M 与向量 $\langle 1, 0, 0 \rangle$, $\langle 0, 1, 0 \rangle$ 和 $\langle 0, 0, 1 \rangle$ 相乘时,可以得到

$$\begin{aligned} \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} &= \begin{bmatrix} M_{11} \\ M_{21} \\ M_{31} \end{bmatrix} \\ \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} &= \begin{bmatrix} M_{12} \\ M_{22} \\ M_{32} \end{bmatrix} \\ \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} &= \begin{bmatrix} M_{13} \\ M_{23} \\ M_{33} \end{bmatrix} \end{aligned} \quad (3.1)$$

类似地, M^{-1} 的列给出了坐标轴从新坐标系到原坐标系的映射。这样,对于任意给定的线性无关的向量 U, V 和 W ,可以构造一个变换矩阵,该矩阵将这些向量映射到向量 $\langle 1, 0, 0 \rangle$, $\langle 0, 1, 0 \rangle$ 和 $\langle 0, 0, 1 \rangle$ 。如下所示,由 U, V 和 W 作为列的矩阵的逆矩阵就具有这种性质:

$$\begin{aligned} \begin{bmatrix} U_x & V_x & W_x \\ U_y & V_y & W_y \\ U_z & V_z & W_z \end{bmatrix}^{-1} \begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix} &= \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} U_x & V_x & W_x \\ U_y & V_y & W_y \\ U_z & V_z & W_z \end{bmatrix}^{-1} \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} &= \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \\ \begin{bmatrix} U_x & V_x & W_x \\ U_y & V_y & W_y \\ U_z & V_z & W_z \end{bmatrix}^{-1} \begin{bmatrix} W_x \\ W_y \\ W_z \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{aligned} \quad (3.2)$$

多个变换可以串连起来,并且可以将多个变换矩阵的乘积用一个矩阵来表示。假设要先用矩阵 M 后用矩阵 G 对一个对象进行变换,由于矩阵乘积满足结合律,对于任意向量 P 都有 $G(MP) = (GM)P$,因此只需存储 GM 的乘积得到的矩阵,将该矩阵作为对象的变换矩阵就可以了。这使我们可以对顶点进行无数次的变换,而不需要额外的存储空间和计算开销。

3.1.1 正交矩阵

正交矩阵是一种其转置矩阵等于其逆矩阵的矩阵。

【定义 3.1】 一个 $n \times n$ 的可逆矩阵 M , 当且仅当 $M^{-1} = M^T$ 时, M 为正交矩阵。

如下面定理所证, 如果矩阵的列形成一个正交向量的集合, 则它是一个正交矩阵。

【定理 3.2】 如果向量组 V_1, V_2, \dots, V_n 构成一个正交向量集合, 那么以 $V_j (1 \leq j \leq n)$ 作为第 j 列的 $n \times n$ 矩阵是正交矩阵。

证明 假设 V_1, V_2, \dots, V_n 构成一个正交向量集合, 令 $n \times n$ 矩阵 M 的列为 V_j 。由于 V_j 是正交的, 则 $V_i \cdot V_j = \delta_{ij}$, 其中 δ_{ij} 是 Kronecker delta 记号。因为矩阵积 $M^T M$ 的 (i, j) 项元素等于点积 $V_i \cdot V_j$, 所以 $M^T M = I$ 。因此, $M^T = M^{-1}$ 。

正交矩阵还有另外一个性质, 就是当用于变换向量时, 正交矩阵不改变向量的长度和角度。对于任意向量 P , 若矩阵 M 不改变其长度, 则有

$$\|MP\| = \|P\| \quad (3.3)$$

对于任意两个向量 P_1 和 P_2 , 若矩阵不改变它们的长度和角度, 可以得到

$$(MP_1) \cdot (MP_2) = P_1 \cdot P_2 \quad (3.4)$$

下面的定理证明了正交矩阵满足式(3.3)和式(3.4)。

【定理 3.3】 如果 $n \times n$ 矩阵 M 是正交的, 那么 M 具有保证长度和角度不变的特性。

证明 设 M 为正交。首先证明用 M 进行变换的向量 P_1 和向量 P_2 的点积不变。然后, 利用这个结论证明 M 不改变长度的特性。考察两个已变换的向量的点积, 可以得到

$$(MP_1) \cdot (MP_2) = (MP_1)^T MP_2 = P_1^T M^T MP_2 \quad (3.5)$$

由于 M 是正交的, 即 $M^{-1} = M^T$, 因此

$$P_1^T M^T MP_2 P_1^T P_2 = P_1 \cdot P_2 \quad (3.6)$$

由于 $\|P\|^2 = P \cdot P$, 这意味着用矩阵 M 进行变换的向量 P 的长度保持不变。

由于正交矩阵不改变长度和角度, 所以它就可以保持坐标系的整体结构。因此, 正交矩阵就只能用于表示旋转和反射的组合。将在 3.3 节中具体讨论旋转。反射指在某一方向上将点进行镜像的一种运算。例如, 矩阵

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (3.7)$$

以 xy 平面为对称面对一点的 z 坐标进行反射。

3.1.2 手向性

在三维空间中, 由 3D 向量 V_1, V_2, V_3 构成的坐标系的基 \mathcal{B} , 具有手向性。对于右手基, 有 $(V_1 \times V_2) \cdot V_3 > 0$ 。也就是说, 在一个右手坐标系中, V_1, V_2 的叉积(遵守右手法则)的方向与 V_3 的方向形成一个锐角。如果 \mathcal{B} 是一个正交规范的右手基, 则有 $V_1 \times V_2 = V_3$ 。若 $(V_1 \times V_2) \cdot V_3 < 0$, 那么 \mathcal{B} 是左手基。

进行奇数次反射操作, 就会改变手向性。偶数次反射相当于一次旋转。所以一系列的反射操作总可以认为是一次简单的旋转加上至多一次反射。通过考察 3×3 矩阵的行列式,

就可以判定矩阵是否存在反射。若 3×3 矩阵 M 的行列式是负的,则存在反射,用 M 对任意基的向量进行变换操作后,基的手向性发生改变。如果行列式是正的,那么 M 不改变手向性。

正交矩阵 M 的行列式的值只可能是 1 或 -1 。若 $\det M = 1$,矩阵 M 只存在旋转;如果 $\det M = -1$,那么 M 表示旋转之后再进行一次反射。

3.2 缩放变换

用 a 作为系数来缩向量 P ,只需计算 $P' = aP$ 。在三维空间中,这个运算也可以表示为矩阵的乘积

$$P' = \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad (3.8)$$

这种缩放称为统一缩放。如图 3.1 所示,如果希望在 x, y 和 z 轴以不同的值缩放向量,那么可以使用与统一缩放矩阵相似的矩阵,只不过其对角线元素不必都相等。称这种缩放为非统一缩放,它可以表示为如下的矩阵乘积

$$P' = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad (3.9)$$

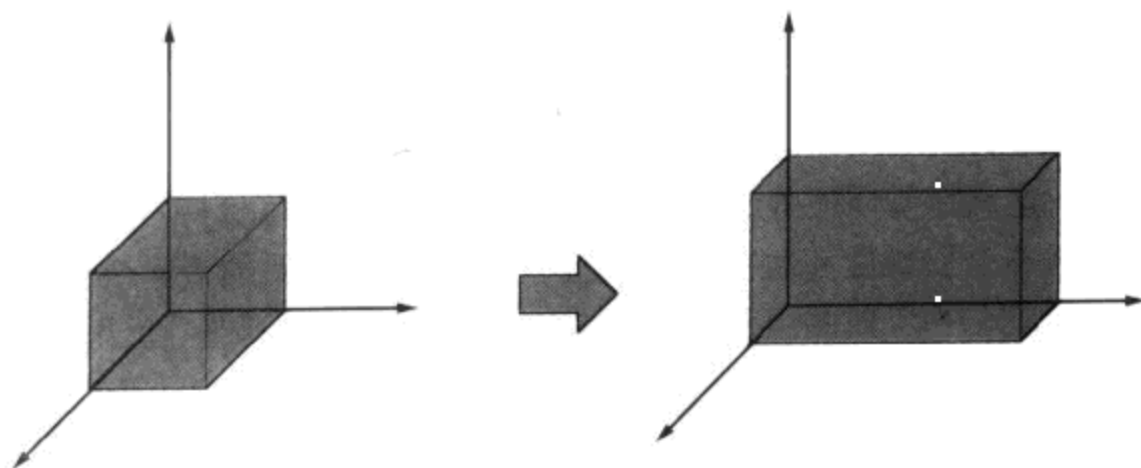


图 3.1 非统一缩放

如果想在 3 个任意轴上进行非统一缩放,就要用到稍微复杂的缩放过程。假设希望以系数 a 沿 U 轴方向,以系数 b 沿 V 轴方向,以系数 c 沿 W 轴方向进行缩放,可以先从坐标系 (U, V, W) 变换到坐标系 (i, j, k) ,然后在 (i, j, k) 坐标系中用等式(3.9)进行缩放运算,最后再还原到 (U, V, W) 坐标系。整个过程可以用下面的矩阵乘积表示

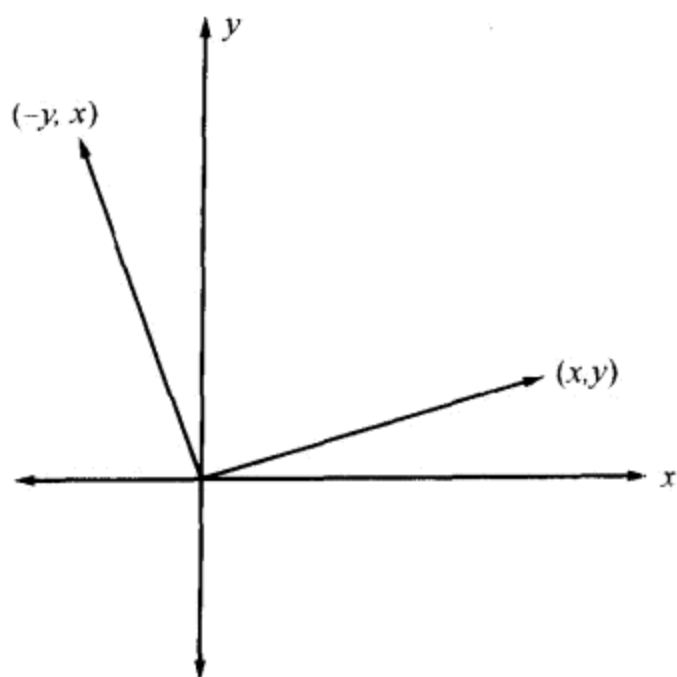
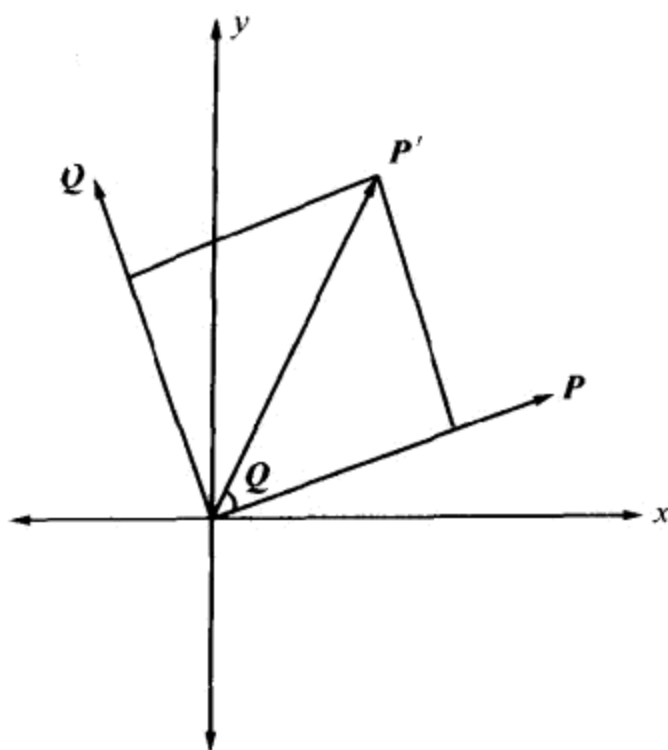
$$P' = \begin{bmatrix} U_x & V_x & W_x \\ U_y & V_y & W_y \\ U_z & V_z & W_z \end{bmatrix} \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} \begin{bmatrix} U_x & V_x & W_x \\ U_y & V_y & W_y \\ U_z & V_z & W_z \end{bmatrix}^{-1} \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad (3.10)$$

3.3 旋转变换

得到将坐标系绕 x, y 或 z 轴旋转 θ 角的 3×3 矩阵并不难。这里对绕 A 轴的正角度旋转是这样规定的:当 A 轴指向我们时,所看到的旋转是逆时针方向。

首先要找到二维空间旋转的通用公式。如图 3.2 所示,通过交换 x 和 y 坐标,并将新的 x 坐标取负,就能够将位于 xy 平面上的二维(2D)向量 P 进行 90° 的逆时针旋转。设旋转向量为 Q ,则有 $Q = \langle -P_y, P_x \rangle$ 。向量 P 和 Q 组成了 xy 平面的一个正交基,因此 xy 平面上的任何向量都可以用这两个向量的线性组合来表示。这里要特别指出,如图 3.3 所示,任意向量 P 以 θ 角旋转后所形成的二维向量 P' ,都可以用分别平行于 P 和 Q 的分量来表示。根据普通三角学有

$$P' = P \cos \theta + Q \sin \theta \quad (3.11)$$

图 3.2 在平面上进行 90° 旋转图 3.3 旋转后向量 P' 的表示

这样,可以用下面的公式得到 P' 分量,即

$$\begin{aligned} P'_x &= P_x \cos \theta - P_y \sin \theta \\ P'_y &= P_y \cos \theta + P_x \sin \theta \end{aligned} \quad (3.12)$$

可以用矩阵的形式改写上面的公式为

$$P' = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} P \quad (3.13)$$

将单位矩阵的第 3 行和第 3 列加入到等式(3.13)中的 2D 旋转矩阵中,就可以将上面的矩阵扩展为绕 z 轴的三维旋转。该 3×3 矩阵可以保证绕 z 轴旋转时,向量的 z 坐标保持不变,这种旋转操作是要经常用到的。这样,绕 z 轴旋转 θ 角度的旋转矩阵 $R_z(\theta)$ 可以表示为

$$R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.14)$$

同样,可以分别得到绕 x 轴和 y 轴旋转 θ 角度的 3×3 旋转矩阵 $R_x(\theta)$ 和 $R_y(\theta)$

$$\begin{aligned} R_x(\theta) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \\ R_y(\theta) &= \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \end{aligned} \quad (3.15)$$

3.3.1 绕任意轴旋转

如果希望将向量 P 绕一个任意轴旋转 θ 角,这里的任意轴以单位向量 A 表示,那么可以先将向量 P 分解为平行于和垂直于 A 的分量,如图 3.4 所示。由于平行分量(P 在 A 上的投影)在旋转过程中保持不变,剩下的问题就是 P 垂直于 A 的分量的旋转。

由于 A 是一个单位向量,对于 P 在 A 上的投影,可以用下面的简化公式来表示:

$$\text{proj}_A P = (A \cdot P)A \quad (3.16) \quad (A \cdot P)A$$

同样,向量 P 垂直于 A 的分量可以表示为

$$\text{perp}_A P = P - (A \cdot P)A \quad (3.17)$$

先将垂直于 A 的分量进行旋转,然后加上由等式 (3.16) 给出的不发生变化的平行分量,就可以得到最终的旋转结果。

垂直分量的旋转是在垂直于 A 轴的平面内进行的。与前面一样,可以用 $\text{perp}_A P$ 与 $\text{perp}_A P$ 逆时针方向旋转 90° 所形成的向量的线性组合来表示旋转后的向量。其实,找到这样的表达式并不难。这里假设 α 是原向量 P 和 A 轴之间的夹角。需要注意的是, $\text{perp}_A P$ 的长度等于 $\|P\| \sin\alpha$, 因为如图 3.4 所示, $\text{perp}_A P$ 可以看作是角 α 的对边,用 $A \times P$ 就可以得到长度为 $\text{perp}_A P$ 并且指向我们所要方向的向量。

现在,用下面的公式可以表示 $\text{perp}_A P$ 旋转 θ 角,即

$$[P - (A \cdot P)A] \cos\theta + (A \times P) \sin\theta \quad (3.18)$$

加上 $\text{perp}_A P$, 就可以得到原向量 P 绕 A 轴旋转的公式为

$$P' = P \cos\theta + (A \times P) \sin\theta + A(A \cdot P)(1 - \cos\theta) \quad (3.19)$$

分别将等式 (3.19) 中 $A \times P$ 和 $A(A \cdot P)$ 替换为公式 (1.25) 和 (1.20) 给出的等价矩阵,得到下面的公式:

$$P' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} P \cos\theta + \begin{bmatrix} 0 & -A_z & A_y \\ A_z & 0 & -A_x \\ -A_y & A_x & 0 \end{bmatrix} P \sin\theta +$$

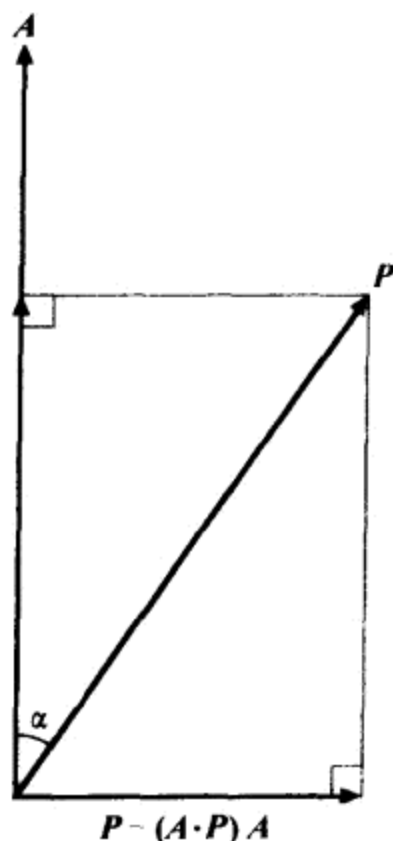


图 3.4 绕任意轴旋转

$$\begin{bmatrix} A_x^2 & A_x A_y & A_x A_z \\ A_x A_y & A_y^2 & A_y A_z \\ A_x A_z & A_y A_z & A_z^2 \end{bmatrix} \mathbf{P}(1 - \cos\theta) \quad (3.20)$$

合并公式中的相同项,并设 $C = \cos\theta$, $S = \sin\theta$,就可以得到表示向量绕 \mathbf{A} 轴旋转 θ 角的旋转矩阵 $\mathbf{R}_A(\theta)$ 。

$$\mathbf{R}_A(\theta) = \begin{bmatrix} C + A_x^2(1 - C) & A_x A_y(1 - C) - A_z S & A_x A_z(1 - C) + A_y S \\ A_x A_y(1 - C) + A_z S & C + A_y^2(1 - C) & A_y A_z(1 - C) - A_x S \\ A_x A_z(1 - C) - A_y S & A_y A_z(1 - C) + A_x S & C + A_z^2(1 - C) \end{bmatrix} \quad (3.21)$$

3.4 齐次坐标

到目前为止,所讨论的变换只涉及到针对三维向量的能够表示成 3×3 矩阵运算的变换。一系列的变换过程最终都可以表示为一个 3×3 矩阵,该矩阵等于各个变换矩阵的乘积。还未讨论一个重要的变换就是平移操作。在空间中平移坐标系,只需简单地添加偏移向量,不会影响坐标轴的方向和尺度。这种运算不能用 3×3 矩阵来表示,把点 \mathbf{P} 从一个坐标系平移到另一个坐标系,通常使用下面的公式:

$$\mathbf{P}' = \mathbf{M}\mathbf{P} + \mathbf{T} \quad (3.22)$$

其中 \mathbf{M} 为 3×3 可逆矩阵, \mathbf{T} 是 3D 平移向量。用公式(3.22)进行两次操作,可以得到比较复杂的结果:

$$\mathbf{P}' = \mathbf{M}_2(\mathbf{M}_1\mathbf{P} + \mathbf{T}_1) + \mathbf{T}_2 = (\mathbf{M}_2\mathbf{M}_1)\mathbf{P} + \mathbf{M}_2\mathbf{T}_1 + \mathbf{T}_2 \quad (3.23)$$

这就要求在连续进行 n 次变换的时候,必须明确地知道每次变换过程中的矩阵分量 $\mathbf{M}_n\mathbf{M}_{n-1}$ 和平移分量 $\mathbf{M}_n\mathbf{T}_{n-1} + \mathbf{T}_n$ 。

3.4.1 四维变换

其实,可以用一种统一的数学形式来简洁而优雅地表示这些变换。为了做到这一点,需要将向量从三维扩展到四维,并使用 4×4 矩阵来表示变换操作。给 3D 点 \mathbf{P} 增加一个坐标,并将这个扩展的第四个坐标(称之为 w 坐标)的值设为 1,这样就将 \mathbf{P} 从三维扩展到四维。这里,我们构造一个 4×4 变换矩阵 \mathbf{F} , \mathbf{F} 对应于 3×3 矩阵 \mathbf{M} 和 3D 变换 \mathbf{T} ,如下所示:

$$\mathbf{F} = \begin{bmatrix} \mathbf{M} & \mathbf{T} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} & T_x \\ M_{21} & M_{22} & M_{23} & T_y \\ M_{31} & M_{32} & M_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.24)$$

将这个矩阵乘以向量 $\langle P_x, P_y, P_z, 1 \rangle$,就以等效于公式(3.22)的方式对向量的 x, y, z 坐标进行变换,同时 w 坐标保持为 1。此外,将形如公式(3.24)的两个矩阵相乘,得到矩阵形式仍符合公式(3.24),并且等效于公式(3.23)中所进行的一对变换。

如果从公式(3.22)解 \mathbf{P} ,则可以得到

$$P = M^{-1}P' - M^{-1}T \quad (3.25)$$

因此,从公式(3.24)中得到 4×4 矩阵 F 的逆矩阵 F^{-1} 为

$$F^{-1} = \left[\begin{array}{c|c} M^{-1} & -M^{-1}T \\ \hline \mathbf{0} & 1 \end{array} \right] = \begin{bmatrix} M_{11}^{-1} & M_{12}^{-1} & M_{13}^{-1} & -(M^{-1}T)_x \\ M_{21}^{-1} & M_{22}^{-1} & M_{23}^{-1} & -(M^{-1}T)_y \\ M_{31}^{-1} & M_{32}^{-1} & M_{33}^{-1} & -(M^{-1}T)_z \\ \hline 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.26)$$

下面的运算验证了上式的正确性。

$$\begin{aligned} FF^{-1} &= \left[\begin{array}{c|c} M & T \\ \hline \mathbf{0} & 1 \end{array} \right] \left[\begin{array}{c|c} M^{-1} & -M^{-1}T \\ \hline \mathbf{0} & 1 \end{array} \right] = \left[\begin{array}{c|c} MM^{-1} & M(-M^{-1}T) + T \\ \hline \mathbf{0} & 1 \end{array} \right] \\ &= \left[\begin{array}{c|c} I_3 & \mathbf{0} \\ \hline \mathbf{0} & 1 \end{array} \right] = I_4 \end{aligned} \quad (3.27)$$

3.4.2 点向量和方向向量

现在必须明确三维空间中表示点的点向量和表示方向的方向向量之间的区别。与点向量不同,方向向量平移过程中将保持不变。如图 3.5 所示,移动坐标系不会影响表示曲线切线的向量的指向。

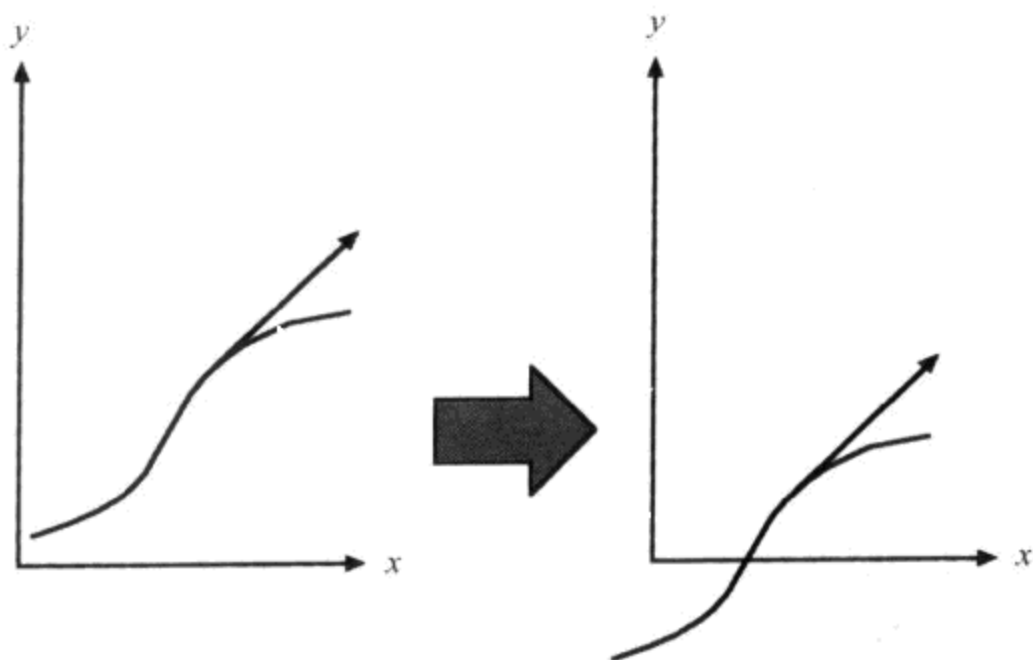


图 3.5 移动坐标系不会改变切线向量的指向

为了能够将变换点向量的 4×4 变换矩阵用于方向向量的变换,必须将方向向量扩展到四维,并将其 w 坐标设为零。在这种情况下,公式(3.24)中的矩阵 F 的第四列就失去了作用,只剩下矩阵左上部的 3×3 部分能够影响方向向量。

计算 P 和 Q 的差可以得到方向向量 $Q - P$,因为 P 和 Q 的 w 坐标都为 1,所以方向向量 $Q - P$ 的 w 坐标为零。这一点不难理解,因为 $Q - P$ 表示的是从 P 指向 Q 的方向,在平移过程中不会发生改变。

3.4.3 坐标的几何解释

关于四维(4D)向量的 w 坐标,迄今为止只用于 4×4 变换矩阵,但它的作用不止于此。前面通过增加值为 1 的 w 坐标将三维的点扩展到了四维,现在要定义一种相反的映射过程。假设有一个四维点 $P = \langle x, y, z, w \rangle$, 其中坐标不为零,这里定义 P 在三维空间的映像,如果用 \tilde{P} 表示 P 在 $w=1$ 的三维空间中的投影, \tilde{P} 可以用下面的公式来表示

$$\tilde{P} = \left\langle \frac{x}{w}, \frac{y}{w}, \frac{z}{w} \right\rangle \quad (3.28)$$

如图 3.6 所示(图中省去 z 轴是为了更直观),三维点 \tilde{P} 对应于连接点 P 与原点的直线和 $w=1$ 的平面的交点。从这一点可以知道,对四维向量 P 作任意系数乘积,其结果都对应于三维空间中一个相同的点。在 4.5 节将详细讨论这种投影在三维图形中的重要性。

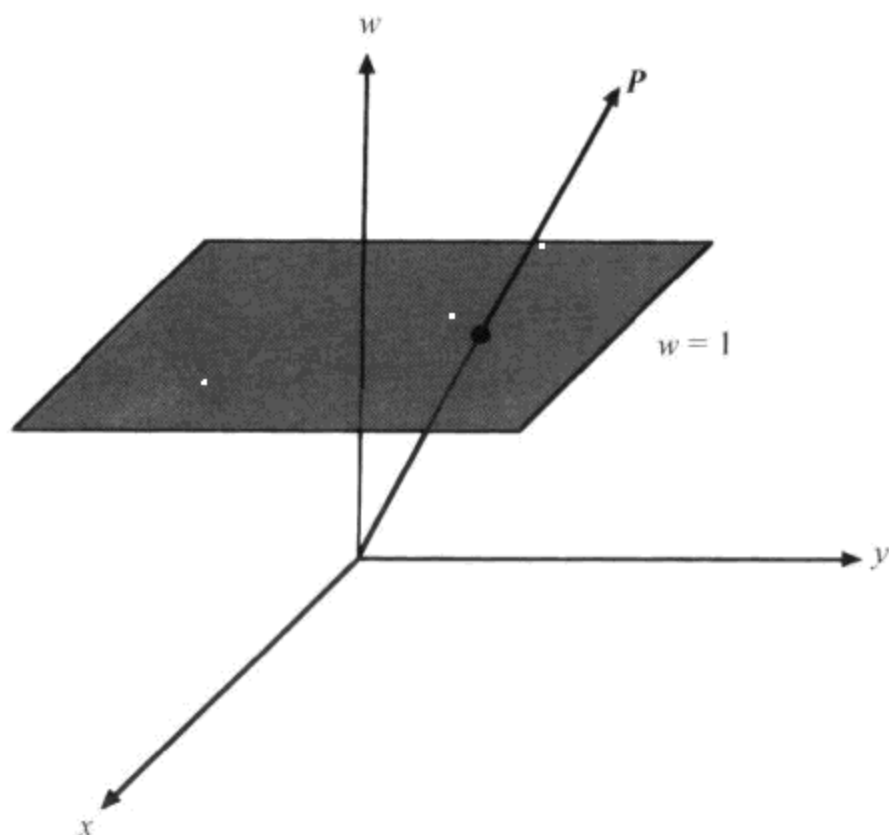
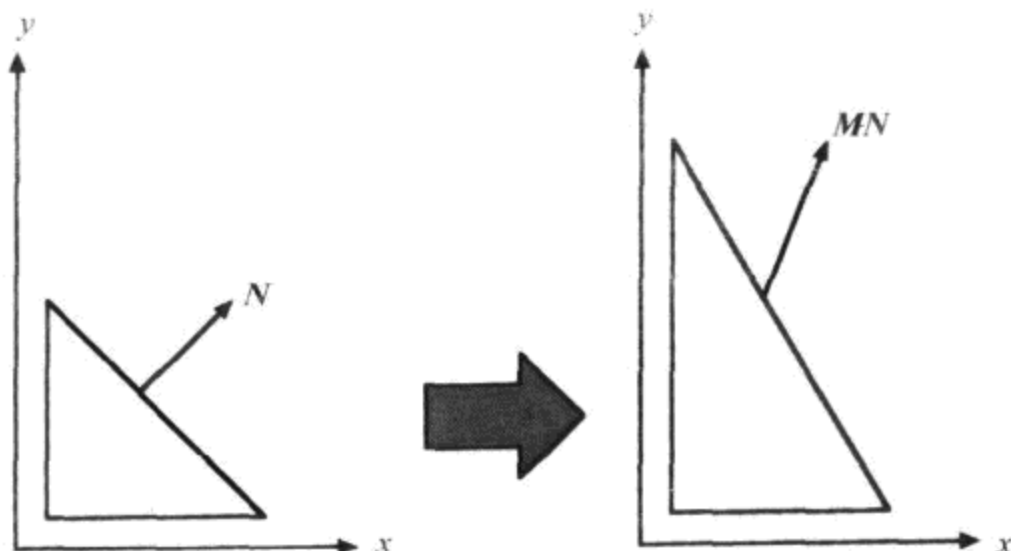


图 3.6 四维点 P 在三维空间的投影

3.5 法向量的变换

一个多边形模型的顶点除了表示空间位置外,还包含一些附加信息,这些附加信息用来表示该顶点是如何融合到周围的曲面中的。一般情况下,顶点可能有切线向量和法线向量与之关联。当变换一个模型时,不但要变换顶点,同时还要对与顶点有关的向量进行变换。

可以通过求两个顶点之间的差来计算切线向量,所以经过变换的切线向量可以表示为两个变换点的差。如果使用 3×3 矩阵 M 变换顶点位置,那么 M 也可以用于变换该顶点的切向量。在这一节,只使用 3×3 矩阵,因为切线方向和法线方向不受平移的影响。在平移法向量时应考虑一些因素。图 3.7 所示为用一个非正交矩阵 M 变换法向量 N 时发生的情况。变换后的法向量最终不垂直于变换后的表面。

图 3.7 用非正交矩阵 M 变换法向量

因为切线方向和法线方向是垂直的,所以同一顶点的切向量 T 和法向量 N 必须满足等式 $N \cdot T = 0$,并且必须保证变换后的切向量 T' 和法向量 N' 仍然能够满足这个等式。设变换矩阵为 M ,有 $T' = MT$ 。这里设变换向量 N 所需的变换矩阵为 G ,可以有

$$N' \cdot T' = (GN) \cdot (MT) = 0 \quad (3.29)$$

经过简单的运算,可以得到

$$(GN) \cdot (MT) = (GN)^T(MT) = N^T G^T M T \quad (3.30)$$

由于 $N^T T = 0$,如果 $G^T M = I$,那么等式 $N^T G^T M T = 0$ 肯定成立。因此,可以得出 $G = (M^{-1})^T$ 。从这里可以看出,利用变换点的矩阵的逆转置矩阵,就能够正确地变换法向量。必须通过这种方式变换的向量称为协变向量。用矩阵 M 以通常的方式进行变换的向量(如点向量和切向量)称为逆变向量。

如果矩阵 M 是正交的,那么 $M^{-1} = M^T$,因此 $(M^{-1})^T = M$ 。所以,当 M 是正交矩阵时,法向量变换过程的求逆转置运算可以避免,这一点适用于本章前面所提到的当 M 等于旋转矩阵 R_x, R_y, R_z 或 R_A 的情况。

3.6 四元数

四元数是三维图像编程人员用来表示旋转的另一种数学形式。在许多情况下用四元数代替旋转矩阵会有更多的好处,因为四元数需要的存储空间更少,四元数之间的连接运算需要的算术运算更少,在产生平滑的三维动画时,用四元数更容易进行修改。

3.6.1 四元数数学

四元数集合,被数学家们称为哈密顿四元数环(ring of Hamiltonian quaternions),用 H 表示,可以理解为四维向量空间,空间中的元素 q 具有以下形式

$$q = [w, x, y, z] = w + xi + yj + zk \quad (3.31)$$

一个四元数经常写成 $q = s + v$ 的形式,其中 s 表示数量部分,对应于 q 的 w 分量, v 表示向量部分,对应于 q 的 x, y, z 分量。

四元数集合是复数集合的自然扩展。四元数的乘法服从分配律,并且虚部 i, j, k 之间

的乘法服从下面的规则

$$\begin{aligned}i^2 &= j^2 = k^2 = -1 \\ij &= -ji = k \\jk &= -kj = i \\ki &= -ik = j\end{aligned}\quad (3.32)$$

四元数式的乘法是不可交换的,因此在运算时应注意正确的顺序。两个四元数 $q_1 = w_1 + x_1i + y_1j + z_1k$ 和 $q_2 = w_2 + x_2i + y_2j + z_2k$ 的乘积 q_1q_2 为

$$\begin{aligned}q_1q_2 &= (w_1w_2 - x_1x_2 - y_1y_2 - z_1z_2) + \\&\quad (w_1x_2 + x_1w_2 + y_1z_2 - z_1y_1)i + \\&\quad (w_1y_2 - x_1z_2 + y_1w_2 + z_1x_2)j + \\&\quad (w_1z_2 + x_1y_2 - y_1x_2 + z_1w_2)k\end{aligned}\quad (3.33)$$

如果写成数量-向量形式,则 $q_1 = s_1 + v_1$ 和 $q_2 = s_2 + v_2$ 的乘积可以写成如下形式

$$q_1q_2 = s_1s_2 - v_1 \cdot v_2 + s_1v_2 + s_2v_1 + v_1 \times v_2 \quad (3.34)$$

和复数(参见附录 A)一样,四元数有共轭。

【定义 3.4】 四元数 $q = s + v$ 的共轭,记为 \bar{q} ,定义为 $\bar{q} = s - v$

通过简单的运算,就可以发现 q 与其共轭 \bar{q} 的乘积等于 q 自身的点积,也等于 q 模的平方。

$$q\bar{q} = \bar{q}q = q \cdot q = \|q\|^2 = q^2 \quad (3.35)$$

这样,就可以导出求四元数乘法逆的公式。

【定理 3.5】 非零的四元数的逆,记为 q^{-1} ,则有

$$q^{-1} = \frac{\bar{q}}{q^2} \quad (3.36)$$

证明 根据等式(3.35),有

$$qq^{-1} = \frac{q\bar{q}}{q^2} = \frac{q^2}{q^2} = 1 \quad (3.37)$$

且

$$q^{-1}q = \frac{\bar{q}q}{q^2} = \frac{q^2}{q^2} = 1 \quad (3.38)$$

由此定理得证。

3.6.2 四元数的旋转

三维空间的旋转可以理解为 R^3 到自身的映射函数 φ 。由于 φ 代表旋转,所以它必须包含有长度、角度、旋转方向等信息。如果有

$$\|\varphi(P)\| = \|P\| \quad (3.39)$$

则长度保持不变。

如果对任意两个点 P_1 和 P_2 有

$$\varphi(P_1) \cdot \varphi(P_2) = P_1 \cdot P_2 \quad (3.40)$$

则从原点到两个点 P_1 和 P_2 的连线所形成的夹角保持不变。

最后,如果

$$\varphi(\mathbf{P}_1) \times \varphi(\mathbf{P}_2) = \varphi(\mathbf{P}_1 \times \mathbf{P}_2) \quad (3.41)$$

则手向性也保持不变。

如果满足条件 $\varphi(s + v) = s + \varphi(v)$, 则函数 φ 可以扩展为 H 到自身的映射, 这样就允许将公式(3.40)重写为

$$\varphi(\mathbf{P}_1) \cdot \varphi(\mathbf{P}_2) = \varphi(\mathbf{P}_1 \cdot \mathbf{P}_2) \quad (3.42)$$

如果将 \mathbf{P}_1 和 \mathbf{P}_2 看成是数量部分为零的四元数, 根据 $\mathbf{P}_1 \mathbf{P}_2 = -\mathbf{P}_1 \cdot \mathbf{P}_2 + \mathbf{P}_1 \times \mathbf{P}_2$, 就可以将等式(3.41)和(3.42)合并成一个等式, 在该等式中可以保持角度不变和手向性不变, 公式如下:

$$\varphi(\mathbf{P}_1) \varphi(\mathbf{P}_2) = \varphi(\mathbf{P}_1 \mathbf{P}_2) \quad (3.43)$$

满足这种等式的函数 φ 称为是同态的。

这一类函数可以用下面的公式给出:

$$\varphi_q(\mathbf{P}) = \mathbf{q} \mathbf{P} \mathbf{q}^{-1} \quad (3.44)$$

其中 \mathbf{q} 为一个非零的四元数, 且满足等式(3.39)和(3.43), 因此可以表示旋转的集合。这一点可以证明。首先可以证明 φ_q 的长度保持不变, 这是因为

$$\|\varphi_q(\mathbf{P})\| = \|\mathbf{q} \mathbf{P} \mathbf{q}^{-1}\| = \|\mathbf{q}\| \|\mathbf{P}\| \|\mathbf{q}^{-1}\| = \|\mathbf{P}\| \frac{\|\mathbf{q}\| \|\bar{\mathbf{q}}\|}{q^2} = \|\mathbf{P}\| \quad (3.45)$$

其次, φ_q 是同态的, 这是因为

$$\varphi_q(\mathbf{P}_1) \varphi_q(\mathbf{P}_2) = \mathbf{q} \mathbf{P}_1 \mathbf{q}^{-1} \mathbf{q} \mathbf{P}_2 \mathbf{q}^{-1} = \mathbf{q} \mathbf{P}_1 \mathbf{P}_2 \mathbf{q}^{-1} = \varphi_q(\mathbf{P}_1 \mathbf{P}_2) \quad (3.46)$$

现在需要找到一个关于四元数 \mathbf{q} 的公式, \mathbf{q} 对应于以 θ 角绕 \mathbf{A} 轴的旋转。简单的计算表明 $\varphi_{a\mathbf{q}} = \varphi_q$ 适用于任何非零数量 a , 为了简单起见, 只涉及单位四元数。设 $\mathbf{q} = s + v$ 是单位四元数, 则 $\mathbf{q}^{-1} = s - v$ 。对于给定的点 \mathbf{P} , 有

$$\begin{aligned} \mathbf{q} \mathbf{P} \mathbf{q}^{-1} &= (s + v) \mathbf{P} (s - v) \\ &= (-v \cdot \mathbf{P} + s \mathbf{P} + v \times \mathbf{P})(s - v) \\ &= -sv \cdot \mathbf{P} + s^2 \mathbf{P} + sv \times \mathbf{P} + (v \cdot \mathbf{P})v - s \mathbf{P} v - (v \times \mathbf{P})v \\ &= s^2 \mathbf{P} + 2sv \times \mathbf{P} + (v \cdot \mathbf{P})v - v \times \mathbf{P} \times v \end{aligned} \quad (3.47)$$

对叉积 $v \times \mathbf{P} \times v$ 使用定理 1.9(6), 这个等式就变为

$$\mathbf{q} \mathbf{P} \mathbf{q}^{-1} = (s^2 - v^2) \mathbf{P} = 2sv \times \mathbf{P} + 2(v \cdot \mathbf{P})v \quad (3.48)$$

设 $v = t\mathbf{A}$, 其中 \mathbf{A} 为单位向量, 则上式可改写为

$$\mathbf{q} \mathbf{P} \mathbf{q}^{-1} = (s^2 - t^2) \mathbf{P} + 2st\mathbf{A} \times \mathbf{P} + 2t^2(\mathbf{A} \cdot \mathbf{P})\mathbf{A} \quad (3.49)$$

比较等式(3.19)给出的绕任意轴旋转的公式, 可以推出下面的等式

$$\begin{aligned} s^2 - t^2 &= \cos\theta \\ 2st &= \sin\theta \\ 2t^2 &= 1 - \cos\theta \end{aligned} \quad (3.50)$$

由第三个等式可以得出

$$t = \sqrt{\frac{1 - \cos\theta}{2}} = \sin \frac{\theta}{2} \quad (3.51)$$

结合式(3.50)中第一和第三个等式, 可以得出 $s^2 + t^2 = 1$, 因此可以肯定 $s = \cos \frac{\theta}{2}$ (由

$\sin 2\theta = 2\sin\theta\cos\theta$ 可以证明 s 和 t 的值满足第二个等式。)

现在可以用下面的等式给出对应于以 θ 角绕 \mathbf{A} 轴旋转的单位四元数 q 为

$$q = \cos \frac{\theta}{2} + \mathbf{A} \sin \frac{\theta}{2} \quad (3.52)$$

这里应注意:四元数 q 的任意数量乘积(特别是 $-q$)表示的是相同的旋转,这是因为

$$(aq)P(aq)^{-1} = aqP\frac{q^{-1}}{a} = qPq^{-1} \quad (3.53)$$

两个四元数 q_1 和 q_2 的乘积也可以表示一个旋转。乘积 q_1q_2 表示先以 q_2 , 后以 q_1 进行旋转。这是因为

$$q_1(q_2Pq_2^{-1})q_1^{-1} = (q_1q_2)P(q_1q_2)^{-1} \quad (3.54)$$

可以将任意多的四元数结合起来,形成表示一系列旋转的一个四元数。将两个四元数相乘需要做 16 次乘法和加法运算,而两个 3×3 矩阵相乘则需要 27 次这样的操作。因此,当对物体进行多次旋转时,应用四元数可以获得较高的计算效率。

经常需要将一个四元数变换成等价的 3×3 旋转矩阵的形式,比如,在需要将一个物体的变换以矩阵的形式传递到一个 3D 图像函数库的时候。利用等式(1.25)和(1.20),可以确定对应于四元数 $q = s + t\mathbf{A}$ 的矩阵的公式,然后将等式(3.49)改写成矩阵形式(这和 3.3.1 节中所用的技巧几乎一样)。由此可以得出

$$qPq^{-1} = \begin{bmatrix} s^2 - t^2 & 0 & 0 \\ 0 & s^2 - t^2 & 0 \\ 0 & 0 & s^2 - t^2 \end{bmatrix} \mathbf{P} + \begin{bmatrix} 0 & -2stA_z & 2stA_y \\ 2stA_z & 0 & -2stA_x \\ -2stA_y & 2stA_x & 0 \end{bmatrix} \mathbf{P} + \begin{bmatrix} 2t^2A_x^2 & 2t^2A_xA_y & 2t^2A_xA_z \\ 2t^2A_xA_y & 2t^2A_y^2 & 2t^2A_yA_z \\ 2t^2A_xA_z & 2t^2A_yA_z & 2t^2A_z^2 \end{bmatrix} \mathbf{P} \quad (3.55)$$

将四元数 q 写成四维向量 $q = \langle w, x, y, z \rangle$, 那么 $w = s, x = tA_x, y = tA_y, z = tA_z$ 。因为 \mathbf{A} 是单位向量,所以 $x^2 + y^2 + z^2 = t^2A^2 = t^2$ 。以 w, x, y 和 z 为分量,则公式(3.55)可改写为

$$qPq^{-1} = \begin{bmatrix} w^2 - x^2 - y^2 - z^2 & 0 & 0 \\ 0 & w^2 - x^2 - y^2 - z^2 & 0 \\ 0 & 0 & w^2 - x^2 - y^2 - z^2 \end{bmatrix} \mathbf{P} + \begin{bmatrix} 0 & -2wx & 2wy \\ 2wx & 0 & -2wz \\ -2wy & 2wz & 0 \end{bmatrix} \mathbf{P} + \begin{bmatrix} 2x^2 & 2xy & 2xz \\ 2xy & 2y^2 & 2yz \\ 2xz & 2yz & 2z^2 \end{bmatrix} \mathbf{P} \quad (3.56)$$

因为 q 是单位四元数,满足 $w^2 + x^2 + y^2 + z^2 = 1$, 所以可以得出

$$w^2 - x^2 - y^2 - z^2 = 1 - 2x^2 - 2y^2 - 2z^2 \quad (3.57)$$

根据该等式,并结合前面的三个矩阵,可以得到用来计算矩阵 \mathbf{R}_q 的公式,这里的 \mathbf{R}_q 表示对应于四元数 q 的旋转矩阵,公式为

$$\mathbf{R}_q = \begin{bmatrix} 1 - 2y^2 - 2z^2 & 2xy - 2wz & 2xz + 2wy \\ 2xy + 2wz & 1 - 2x^2 - 2z^2 & 2yz - 2wx \\ 2xz - 2wy & 2yz + 2wx & 1 - 2x^2 - 2y^2 \end{bmatrix} \quad (3.58)$$

3.6.3 球型线性插值

因为四元数是用向量表示的,所以很适合于插值运算。在产生一个物体动画的过程中,在产生位于两个预先计算的关键帧之间的中间过渡定位时,插值非常有用。

最简单的插值类型是线性插值。对于两个四元数 q_1 和 q_2 ,线性插值后所得的四元数 $q(t)$ 为

$$q(t) = (1-t)q_1 + tq_2 \quad (3.59)$$

当 t 在 0 和 1 之间变化时,函数 $q(t)$ 在连接 q_1 和 q_2 的线段上平滑变化。如图 3.8 所示, $q(t)$ 并不保持 q_1 和 q_2 的单位长度,但可以使用下面的函数在任意点位置对 $q(t)$ 进行重新规格化:

$$q(t) = \frac{(1-t)q_1 + tq_2}{\|(1-t)q_1 + tq_2\|} \quad (3.60)$$

这样,可以用该函数来描绘位于 q_1 和 q_2 间的过渡弧线,如图 3.8 所示。它将弧描绘成四维单位超球面的二维截面。

尽管线性插值是很有效的,但是由等式(3.60)得出的函数 $q(t)$ 并没有以恒定的速率描绘 q_1 和 q_2 间的过渡弧线,这是线性插值的弊端。图 3.9 中关于 $\cos^{-1}(q(t) \cdot q_1)$ 的图形表明, q_1 和 q_2 之间的角度变化速率在端点 $t=0$ 和 $t=1$ 时相对较慢,而在 $t=1/2$ 时最快。

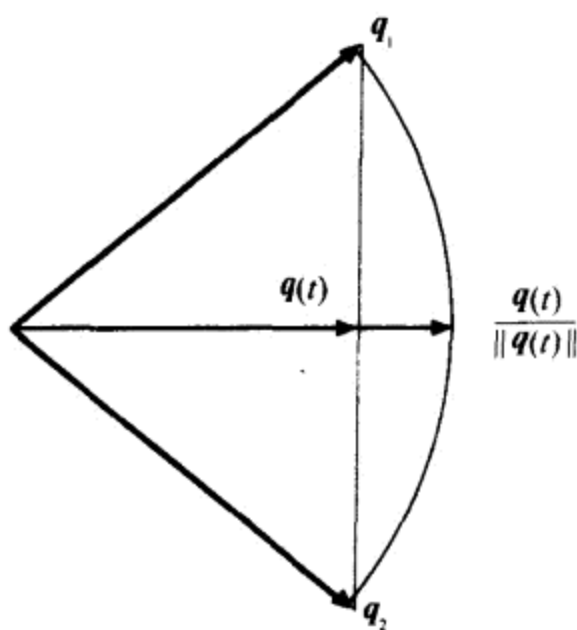


图 3.8 四元数的线性插值

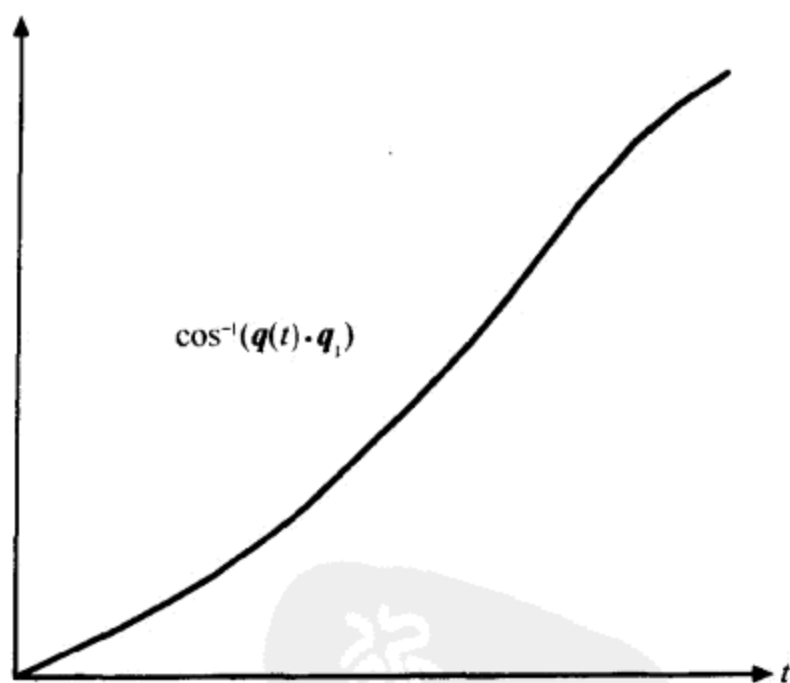


图 3.9 $\cos^{-1}(q(t) \cdot q_1)$ 的曲线($q(t)$ 是等式(3.60)给出的规格化线性插值函数)

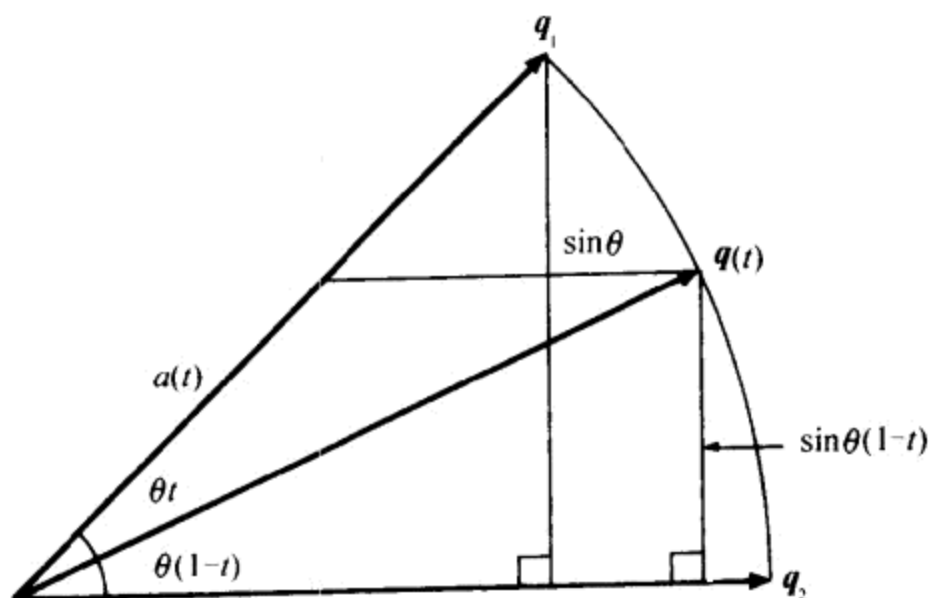
我们希望找到一个函数 $q(t)$, 用它对四元数 q_1 和 q_2 进行插值时, 会保持其单位长度不变并且以恒定的速率扫过位于 q_1 和 q_2 之间的夹角。如果 q_1 和 q_2 的夹角为 θ , 那么这

个函数将会产生一个四元数,该四元数在 $q(t)$ 和 q_1 之间形成一个夹角 θt , 这里 t 在 0 到 1 之间取值。

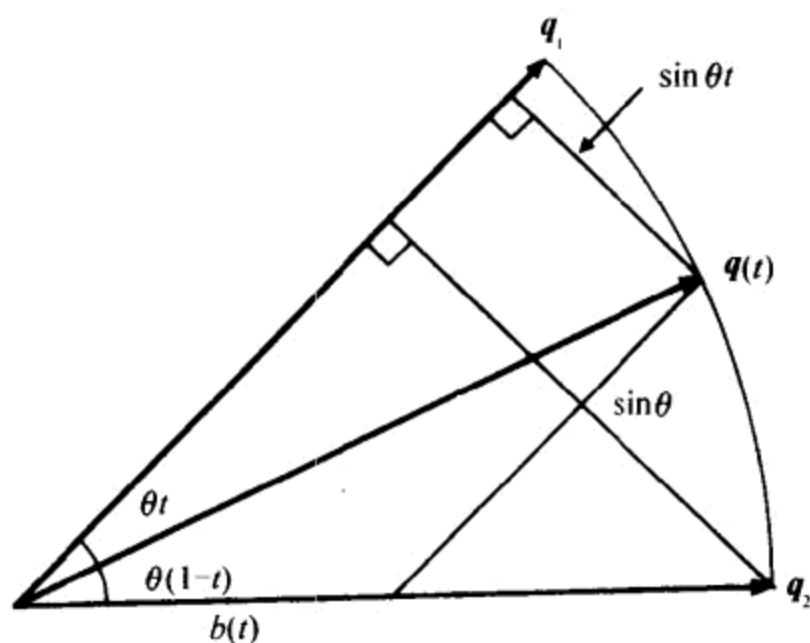
如图 3.10 所示,四元数 $q(t)$ 位于连接 q_1 和 q_2 的弧上,与 q_1 构成夹角 θt , 与 q_2 构成夹角 $\theta(1-t)$ 。可以将 $q(t)$ 写成

$$q(t) = a(t)q_1 + b(t)q_2 \quad (3.61)$$

其中 $a(t)$ 和 $b(t)$ 分别表示 $q(t)$ 在 q_1 和 q_2 方向上的分量的长度。



(a) $q(t)$ 沿 q_1 方向的分量



(b) $q(t)$ 沿 q_2 方向的分量

图 3.10 用相似三角形得到分量长度

如图 3.10(a)所示,可以构造相似三角形来确定长度 $a(t)$ 。 q_1 到以原点和 q_2 为端点的线段的垂直距离为 $\|q_1\| \sin\theta$, 而 $q(t)$ 到该线段的垂直距离为 $\|q(t)\| \sin\theta(1-t)$ 。根据相似三角形,可以得到以下关系式:

$$\frac{a(t)}{\|q_1\|} = \frac{\|q(t)\| \sin\theta(1-t)}{\|q_1\| \sin\theta} \quad (3.62)$$

由于 $\|q_1\| = 1, \|q(t)\| = 1$, 可以将上式简化为

$$a(t) = \frac{\sin\theta(1-t)}{\sin\theta} \quad (3.63)$$

图 3.10(b) 表示了求长度 $b(t)$ 的相同过程, 这里有

$$b(t) = \frac{\sin\theta t}{\sin\theta} \quad (3.64)$$

现在可以将球型线性插值函数 $q(t)$ 定义如下

$$q(t) = \frac{\sin\theta(1-t)}{\sin\theta} q_1 + \frac{\sin\theta t}{\sin\theta} q_2 \quad (3.65)$$

这里的角 θ 为

$$\theta = \cos^{-1}(q_1 \cdot q_2) \quad (3.66)$$

如果需要, $\sin\theta$ 可以用下式替换, 即

$$\sin\theta = \sqrt{1 - (q_1 \cdot q_2)^2} \quad (3.67)$$

因为四元数 q 和 $-q$ 表示相同的旋转, 所以在选择四元数 q_1 和 q_2 的正负号时一般要满足 $q_1 \cdot q_2 \geq 0$, 这样也可以保证以最短路径的方式进行插值。

3.7 本章小结

正交矩阵

对于 $n \times n$ 可逆矩阵 M , 如果满足 $M^{-1} = M^T$, 则称 M 为正交矩阵。如果矩阵的列所组成的向量组是线性无关的, 则该矩阵是正交的。正交矩阵保证长度和角度不变, 因此只能用于旋转和反射。

缩放变换

三维空间的缩放运算通过下面的变换矩阵来实现:

$$\begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix}$$

如果 $a = b = c$, 则该矩阵表示的是统一缩放, 统一缩放也可以通过数量乘法来实现。

旋转变换

以角度 θ 绕 x, y 和 z 轴旋转的运算通过以下变换矩阵实现:

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

以角度 θ 绕任意轴 \mathbf{A} 旋转的运算通过下面的变换矩阵来实现:

$$\mathbf{R}_A(\theta) = \begin{bmatrix} C + A_x^2(1-C) & A_x A_y(1-C) - A_z S & A_x A_z(1-C) + A_y S \\ A_x A_y(1-C) + A_z S & C + A_y^2(1-C) & A_y A_z(1-C) - A_x S \\ A_x A_z(1-C) - A_y S & A_y A_z(1-C) + A_x S & C + A_z^2(1-C) \end{bmatrix}$$

其中 $C = \cos\theta$, $S = \sin\theta$ 。

齐次坐标

向量 \mathbf{P} 表示一个三维点,可以通过增加一个值为 1 的 w 坐标而扩展成四维的齐次坐标。向量 \mathbf{D} 表示一个三维方向,可以通过增加一个值为 0 的 w 坐标而扩展成四维的齐次坐标。

对于 3×3 变换矩阵 \mathbf{M} 和 3D 平移向量 \mathbf{T} ,可以利用下面的 4×4 变换矩阵将它们组合到一起:

$$\mathbf{F} = \begin{bmatrix} M_{11} & M_{12} & M_{13} & T_x \\ M_{21} & M_{22} & M_{23} & T_y \\ M_{31} & M_{32} & M_{33} & T_z \\ \hline 0 & 0 & 0 & 1 \end{bmatrix}$$

变换法向量时,必须使用变换点的矩阵的逆转置矩阵。

四元数

对应于绕单位轴 \mathbf{A} 旋转 θ 角度的单位四元数满足

$$\mathbf{q} = \cos \frac{\theta}{2} + \mathbf{A} \sin \frac{\theta}{2}$$

四元数 \mathbf{q} 通过相似矩阵 $\mathbf{P}' = \mathbf{qPq}^{-1}$ 对点 \mathbf{P} 进行旋转变换。四元数 $\mathbf{q} = \langle w, x, y, z \rangle$ 能完成的变换等价于下面 3×3 矩阵完成的变换

$$\mathbf{R}_q = \begin{bmatrix} 1 - 2y^2 - 2z^2 & 2xy - 2wz & 2xz + 2wy \\ 2xy + 2wz & 1 - 2x^2 - 2z^2 & 2yz - 2wx \\ 2xz - 2wy & 2yz + 2wx & 1 - 2x^2 - 2y^2 \end{bmatrix} \quad (3.68)$$

球型线性插值

两个四元数 \mathbf{q}_1 和 \mathbf{q}_2 可以通过下面的公式进行球形插值:

$$\mathbf{q}(t) = \frac{\sin\theta(1-t)}{\sin\theta} \mathbf{q}_1 + \frac{\sin\theta t}{\sin\theta} \mathbf{q}_2$$

其中 $0 \leq t \leq 1$ 。

习 题

1. 计算绕 x 、 y 和 z 轴旋转 30° 的 3×3 旋转矩阵。
2. 写出以 60° 角绕轴 $\langle 0, 3, 4 \rangle$ 旋转的单位四元数。
3. 证明等式(3.34)。
4. 设 N 是表面在点 P 处的法向量, S 和 T 为点 P 处的切向量, 使得 $S \times T = N$ 。对于给定 3×3 可逆矩阵 M , 假设法向量已经通过矩阵 M 的逆转置正确地进行变换, 试证明: $(MS) \times (\det M)(M^{-1})^T(S \times T)$ 。

[提示: 利用等式(1.25)将叉积 $(MS) \times (MT)$ 写成

$$(MS) \times (MT) = \begin{bmatrix} 0 & -(MS)_z & (MS)_y \\ (MS)_z & 0 & -(MS)_x \\ -(MS)_y & (MS)_x & 0 \end{bmatrix} MT$$

然后, 找出矩阵 G :

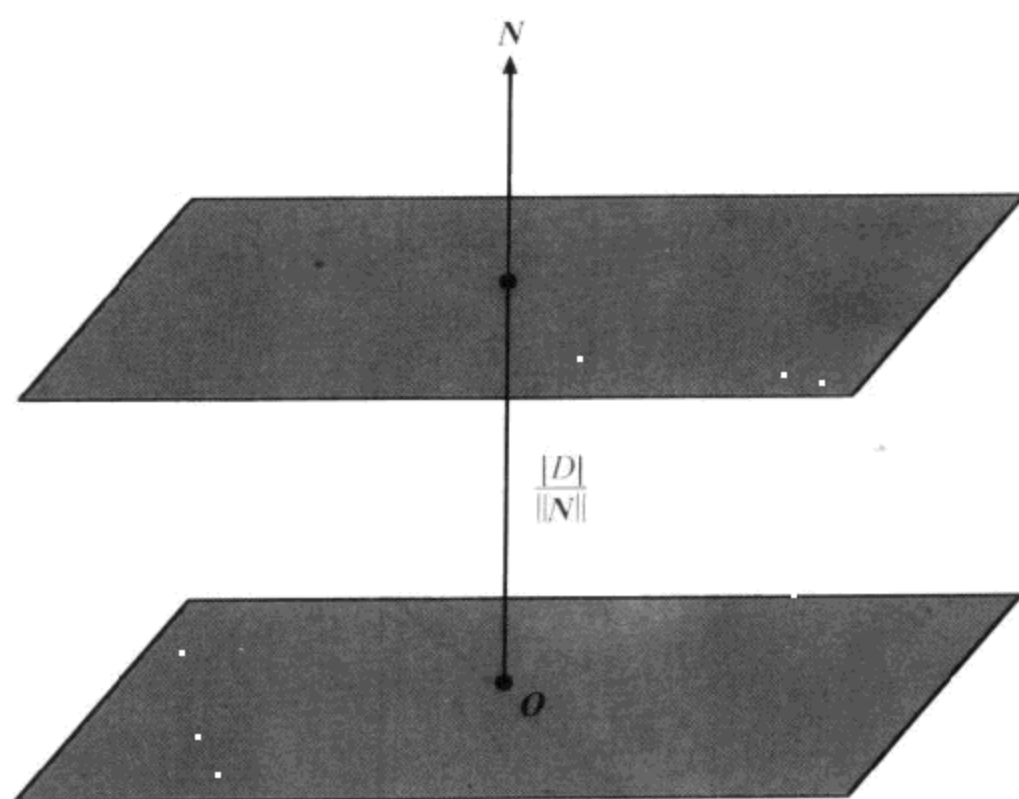
$$G \begin{bmatrix} 0 & -S_z & S_y \\ S_z & 0 & -S_x \\ -S_y & S_x & 0 \end{bmatrix} = \begin{bmatrix} 0 & -(MS)_z & (MS)_y \\ (MS)_z & 0 & -(MS)_x \\ -(MS)_y & (MS)_x & 0 \end{bmatrix} M$$

最后应用等式(2.64)证明 $G = (\det M)(M^{-1})^T$

5. 用 C++ 语言实现一个封装四元数的类。该类必须包含用以描述四元数的 w, x, y 和 z 分量的数据成员。除了具有一个不进行任何初始化操作的默认构造函数之外, 该类还需要一个构造函数, 该构造函数以 4 个浮点数作为参数, 并将参数值赋于 4 个分量作为初始化。该类还需要重载对数量系数的加、减、乘、除运算符, 以及等式(3.33)定义的四元数积。该类应该拥有一个成员函数, 以 θ 和轴 A 作为输入参数, 返回表示绕 A 旋转 θ 角的单位四元数。另外, 该类应该包括一个根据等式(3.58)将四元数转化为 3×3 旋转矩阵的成员函数。最后, 类还应该包括计算四元数的模和求四元数的逆的函数。



第 4 章 三维引擎中的几何



在这一章中,开始学习如何将前 3 章讲到的知识应用到具有艺术性和科学性的 3D 游戏编程中去。首先学习三维空间中的直线和平面的特性,之后介绍视截体,同时还要介绍在操纵虚拟摄像机(我们通过它来观察整个游戏世界)的过程中用到的一些重要数学方法。

4.1 3D 空间中的直线

对于给定两个 3D 点 P_1 和 P_2 ,可以把通过这两个点的直线定义为

$$P(t) = (1 - t)P_1 + tP_2 \quad (4.1)$$

其中, t 可以是任意实数。 P_1 与 P_2 之间的线段对应于 t 在 0 到 1 之间的值。

一条射线是指只有一个端点并且在给定的方向 V 上无限延伸的直线。射线通常表示为参数方程

$$P(t) = P_0 + tV \quad (4.2)$$

其中, t 的值大于等于 0。这个方程通常也可以用来表示直线,如果令 $P_0 = P_1$, $V = P_2 - P_1$,则这个等式与等式(4.1)等价。

4.1.1 点到直线的距离

如图 4.1 所示,给定一个点 Q ,给定的直线由端点 P_0 和方向向量 V 定义,在计算 Q 到直线的距离 d 时,可以通过计算 $Q - P_0$ 相对于直线的垂直分量而得到。

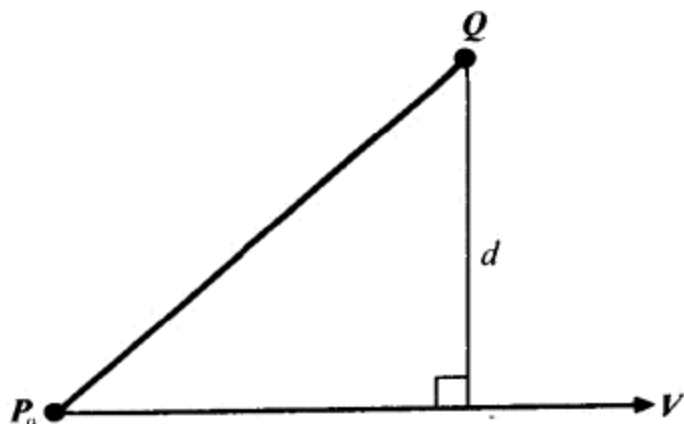


图 4.1 计算点到直线的距离

根据毕达哥拉斯(Pythagorean)定理,从点 Q 到直线的距离的平方等于 $Q - P_0$ 的平方减去 $Q - P_0$ 在 V 方向上投影的平方,如下式所示:

$$\begin{aligned} d^2 &= (Q - P_0)^2 - [\text{proj}_D(Q - P_0)]^2 \\ &= (Q - P_0)^2 - \left[\frac{(Q - P_0) \cdot V}{V^2} V \right]^2 \\ &= (Q - P_0)^2 - \frac{[(Q - P_0) \cdot V]^2}{V^2} \end{aligned} \quad (4.3)$$

求结果的平方根就能得到所需的距离 d 。

4.1.2 两条直线之间的距离

在二维空间中,两直线要么平行要么相交于一点。但在三维空间中,这种说法是不成立

的。如果两条直线既不平行也不相交,那么称这两条直线异面。求异面直线上点间最短距离的公式时,其过程并不难。

假定有两条直线,如图 4.2 所示,由如下参数方程表示

$$\begin{aligned} P(s) &= P_0 + sV_P \\ Q(t) &= Q_0 + tV_Q \end{aligned} \quad (4.4)$$

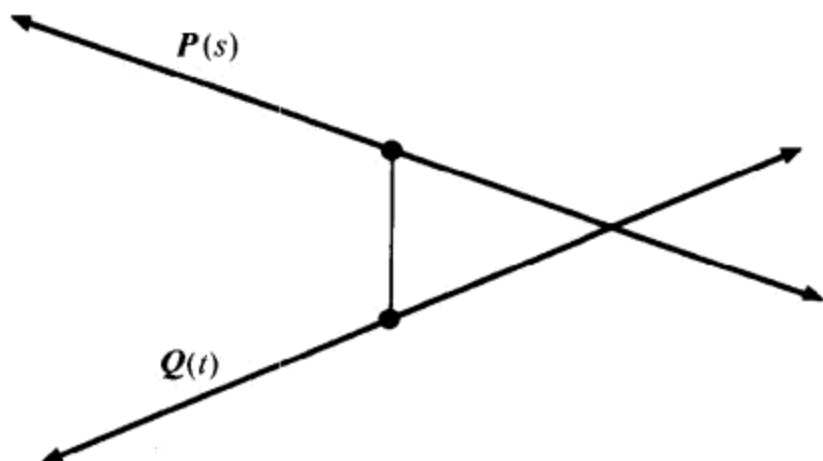


图 4.2 计算异面直线间的距离

其中 s 和 t 可以取任意实数。从直线 $P(s)$ 上的一点到直线 $Q(t)$ 上一点的距离的平方可以表示为参数为 s 和 t 的函数

$$f(s, t) = \|P(s) - Q(t)\|^2 \quad (4.5)$$

将平方展开,并将函数 $P_0(s)$ 和 $Q(t)$ 代入,可得

$$\begin{aligned} f(s, t) &= P(s)^2 + Q(t)^2 - 2P(s) \cdot Q(t) \\ &= (P_0 + sV_P)^2 + (Q_0 + tV_Q)^2 - \\ &\quad 2P_0 \cdot Q_0 - 2sV_P \cdot Q_0 - 2tV_Q \cdot P_0 - 2stV_P \cdot V_Q \\ &= P_0^2 + s^2V_P^2 + 2sP_0 \cdot V_P + Q_0^2 + t^2V_Q^2 + 2tQ_0 \cdot V_Q - \\ &\quad 2P_0 \cdot Q_0 - 2sV_P \cdot Q_0 - 2tV_Q \cdot P_0 - 2stV_P \cdot V_Q \end{aligned} \quad (4.6)$$

分别求函数 f 关于 s 和 t 的偏导数,并使偏导数为零,这样就可以求得 f 的最小值为

$$\frac{\partial f}{\partial s} = 2sV_P^2 + 2P_0 \cdot V_P - 2V_P \cdot Q_0 - 2tV_P \cdot V_Q = 0 \quad (4.7)$$

$$\frac{\partial f}{\partial t} = 2tV_Q^2 + 2Q_0 \cdot V_Q - 2V_Q \cdot P_0 - 2sV_P \cdot V_Q = 0 \quad (4.8)$$

去掉系数 2 后,可以将上面的两个式子用下面的矩阵形式来表示:

$$\begin{bmatrix} V_P^2 & -V_P \cdot V_Q \\ -V_P \cdot V_Q & V_Q^2 \end{bmatrix} \begin{bmatrix} s \\ t \end{bmatrix} = \begin{bmatrix} (Q_0 - P_0) \cdot V_P \\ (P_0 - Q_0) \cdot V_Q \end{bmatrix} \quad (4.9)$$

解关于 s 和 t 的方程,可得

$$\begin{aligned} \begin{bmatrix} s \\ t \end{bmatrix} &= \begin{bmatrix} V_P^2 & -V_P \cdot V_Q \\ -V_P \cdot V_Q & V_Q^2 \end{bmatrix}^{-1} \begin{bmatrix} (Q_0 - P_0) \cdot V_P \\ (P_0 - Q_0) \cdot V_Q \end{bmatrix} \\ &= \frac{1}{V_P^2V_Q^2 - (V_P \cdot V_Q)^2} \begin{bmatrix} V_Q^2 & V_P \cdot V_Q \\ V_P \cdot V_Q & V_P^2 \end{bmatrix} \begin{bmatrix} (Q_0 - P_0) \cdot V_P \\ (P_0 - Q_0) \cdot V_Q \end{bmatrix} \end{aligned} \quad (4.10)$$

把 s 和 t 的值代回函数 f , 可以得到两直线间距离平方的最小值, 开平方后就得到所求的直线间距离。如果方向向量 V_P 和 V_Q 是单位向量, 则有 $V_P^2 = 1, V_Q^2 = 1$, 那么等式 (4.10) 就会更为简单些。

如果 $V_P^2 V_Q^2 - (V_P \cdot V_Q)^2$ 的值等于零, 那么两条直线平行, 在这种情况下, 两条直线的距离等于任意一条直线上的点到另一条直线的距离, 如图 4.3 所示。当然, 可以用等式 (4.3) 来计算从点 P_0 到直线 $Q(t)$ 或者从点 Q_0 到直线 $P(s)$ 的距离。

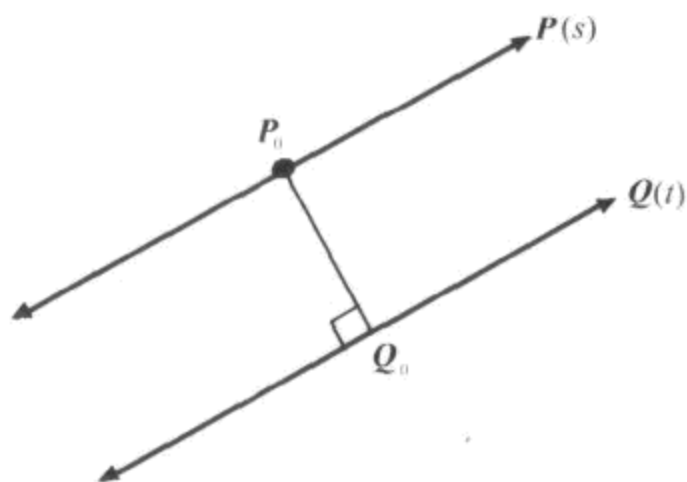


图 4.3 计算两条平行直线之间的距离

4.2 3D 空间中的平面

对于给定的 3D 点 P_0 和法向量 N , 那么过 P_0 且与 N 垂直的平面可以定义为满足方程 $N \cdot (P - P_0) = 0$ 的点 P 的集合。如图 4.4 所示, 平面中任意点与 P_0 的连线都与法线 N 垂直。

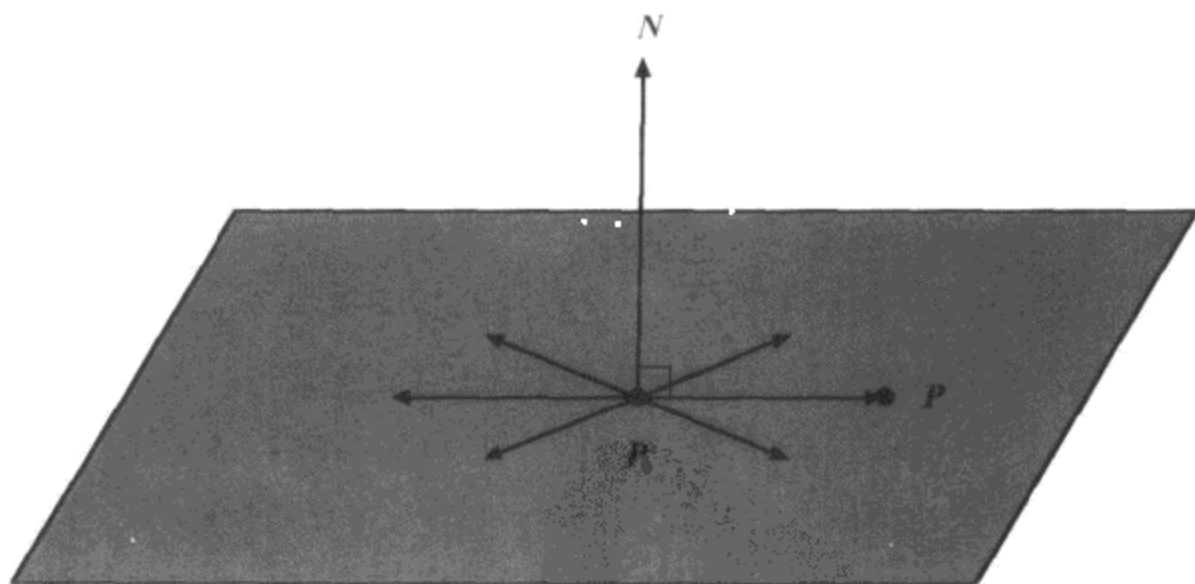


图 4.4 P 与平面上已知的点 P_0 的连线都与法线 N 垂直

平面方程通常可以表示为

$$Ax + By + Cz + D = 0 \quad (4.11)$$

其中 A, B 和 C 是法向量 N 的 x, y 和 z 分量, $D = -N \cdot P_0$ 。如图 4.5 所示, $|D| / \|N\|$ 是该

平面和经过原点且与该平面平行的平面之间的距离。

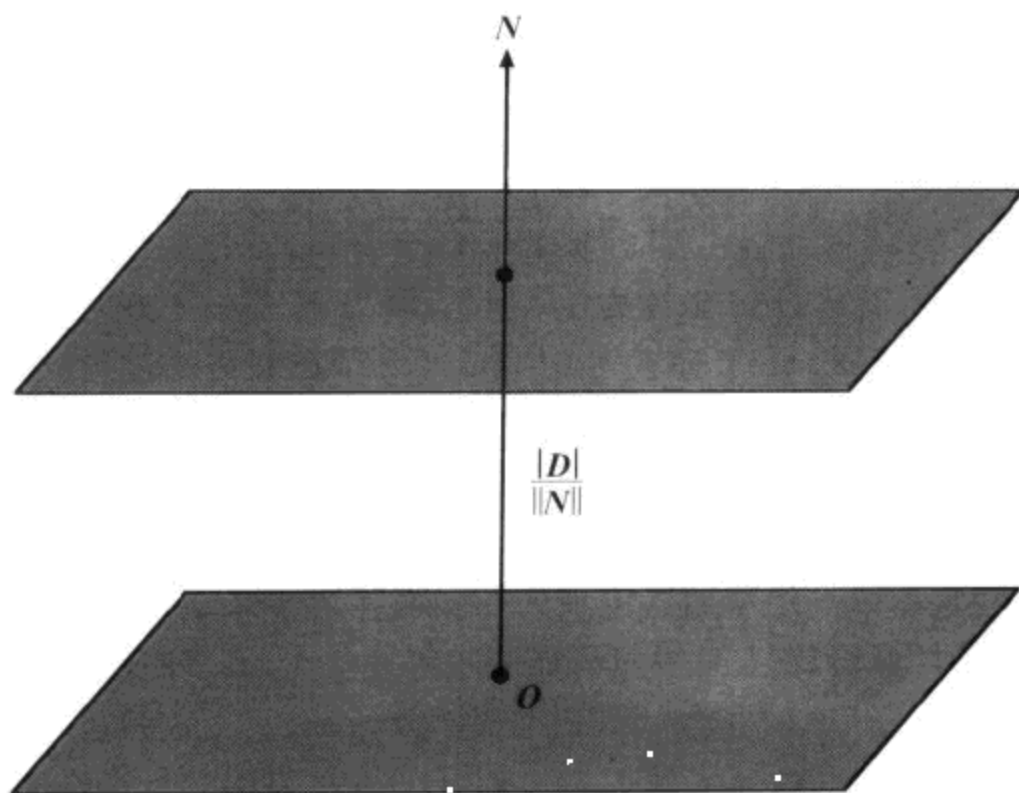


图 4.5 等式(4.11)中的 D 值与原点 to 该平面的垂直距离成比例

法向量 \mathbf{N} 经常被规格化为单位长度,因为这样等式

$$d = \mathbf{N} \cdot \mathbf{Q} + D \quad (4.12)$$

就可以表示平面到任意点 \mathbf{Q} 的带符号距离。如果 $d=0$,则点 \mathbf{Q} 就在平面上。如果 $D>0$,则称点 \mathbf{Q} 在平面的正侧,因为此时 \mathbf{Q} 位于平面法向量所指的一侧。与此相反,如果 $d<0$,那么 \mathbf{Q} 位于平面的负侧。

用一个四维向量可以方便地表示一个平面。缩写记号 $\langle \mathbf{N}, D \rangle$ 用来表示满足方程 $\mathbf{N} \cdot \mathbf{Q} + D=0$ 的点 \mathbf{Q} 所组成的平面。如果将三维的点用 w 坐标为 1 的四维齐次坐标点来表示,那么等式(4.12)可以写成 $d = \mathbf{L} \cdot \mathbf{Q}$,其中 $\mathbf{L} = \langle \mathbf{N}, D \rangle$ 。如果 $\mathbf{L} \cdot \mathbf{Q} = 0$,那么点 \mathbf{Q} 位于平面上。

4.2.1 直线与平面相交

确定直线与平面的交点,是 3D 引擎中的一种常用计算。特别是在计算多边形裁剪的过程中会大量地用到直线与平面的求交,有关具体的讨论详见 7.4.1 和 9.2.2 节。

设 $\mathbf{P}(t) = \mathbf{Q} + t\mathbf{V}$ 表示包含点 \mathbf{Q} 并沿 \mathbf{V} 方向延伸的直线。对于法向量为 \mathbf{N} 并且距原点带符号距离为 D 的平面,可以通过解下面的方程来确定直线与平面的交点

$$\mathbf{N} \cdot \mathbf{P}(t) + D = 0 \quad (4.13)$$

用 $\mathbf{Q} + t\mathbf{V}$ 替代 $\mathbf{P}(t)$,可以得到

$$\mathbf{N} \cdot \mathbf{Q} + (\mathbf{N} \cdot \mathbf{V})t + D = 0 \quad (4.14)$$

解关于 t 的方程,可得

$$t = \frac{-\mathbf{N} \cdot \mathbf{Q} - D}{\mathbf{N} \cdot \mathbf{V}} \quad (4.15)$$

把 t 的值代回直线方程 $\mathbf{P}(t) = \mathbf{Q} + t\mathbf{V}$,就可以求得直线与平面的交点。如果 $\mathbf{N} \cdot \mathbf{V}$ 为零,

那么直线与平面平行(平面法向与直线方向 \mathbf{V} 垂直)。在这种情况下,如果 $\mathbf{N} \cdot \mathbf{Q} + D = 0$,那么直线在该平面上;否则,直线和平面没有交点。

我们也可以用平面的四维表示法来给出等式(4.15)中的 t 值。对于给定平面 $L = \langle \mathbf{N}, D \rangle$,可以得到

$$t = -\frac{\mathbf{L} \cdot \mathbf{Q}}{\mathbf{L} \cdot \mathbf{V}} \quad (4.16)$$

因为 \mathbf{Q} 是一个点向量,所以它的 w 坐标为 1。但是,因为 \mathbf{V} 是一个方向向量,所以在将它扩展成齐次坐标时需要将它的 w 坐标置为 0(在 3.4.2 节已经讨论过)。这也证明了等式(4.16)与等式(4.15)是等价的。

4.2.2 三个平面相交

空间中的区域经常用由一组平面所围成的凸多面体边界来定义。凸多面体的顶点和棱,可以通过一系列的计算得到,在计算过程需要求解一些点,有多组的三个平面在这些点相交。

设 $L_1 = \langle \mathbf{N}_1, D_1 \rangle$, $L_2 = \langle \mathbf{N}_2, D_2 \rangle$ 及 $L_3 = \langle \mathbf{N}_3, D_3 \rangle$ 表示任意三个平面,可以通过求解下面的方程组来得到同时位于这三个平面的点 \mathbf{Q} :

$$\begin{aligned} \mathbf{L}_1 \cdot \mathbf{Q} &= 0 \\ \mathbf{L}_2 \cdot \mathbf{Q} &= 0 \\ \mathbf{L}_3 \cdot \mathbf{Q} &= 0 \end{aligned} \quad (4.17)$$

可以写成矩阵的形式

$$\mathbf{M}\mathbf{Q} = \begin{bmatrix} -D_1 \\ -D_2 \\ -D_3 \end{bmatrix} \quad (4.18)$$

其中,矩阵 \mathbf{M} 为

$$\mathbf{M} = \begin{bmatrix} (\mathbf{N}_1)_x & (\mathbf{N}_1)_y & (\mathbf{N}_1)_z \\ (\mathbf{N}_2)_x & (\mathbf{N}_2)_y & (\mathbf{N}_2)_z \\ (\mathbf{N}_3)_x & (\mathbf{N}_3)_y & (\mathbf{N}_3)_z \end{bmatrix} \quad (4.19)$$

假设矩阵 \mathbf{M} 是可逆的,解出点 \mathbf{Q} ,就可以得到三个平面的惟一相交点,求解方程如下

$$\mathbf{Q} = \mathbf{M}^{-1} \begin{bmatrix} -D_1 \\ -D_2 \\ -D_3 \end{bmatrix} \quad (4.20)$$

如果 \mathbf{M} 是奇异矩阵(即 $\det \mathbf{M} = 0$),那么这三个平面不相交于一点。当三个平面的法线位于同一个平面时,会出现这种情况,这样的例子如图 4.6 所示。

当两个相互不平行的面 $L_1 = \langle \mathbf{N}_1, D_1 \rangle$ 与 $L_2 = \langle \mathbf{N}_2, D_2 \rangle$ 相交时,它们相交于一直线。如图 4.7 所示,交线的方向 \mathbf{V} ,分别与两个平面的法线垂直,因此可表示为 $\mathbf{V} = \mathbf{N}_1 \times \mathbf{N}_2$ 。

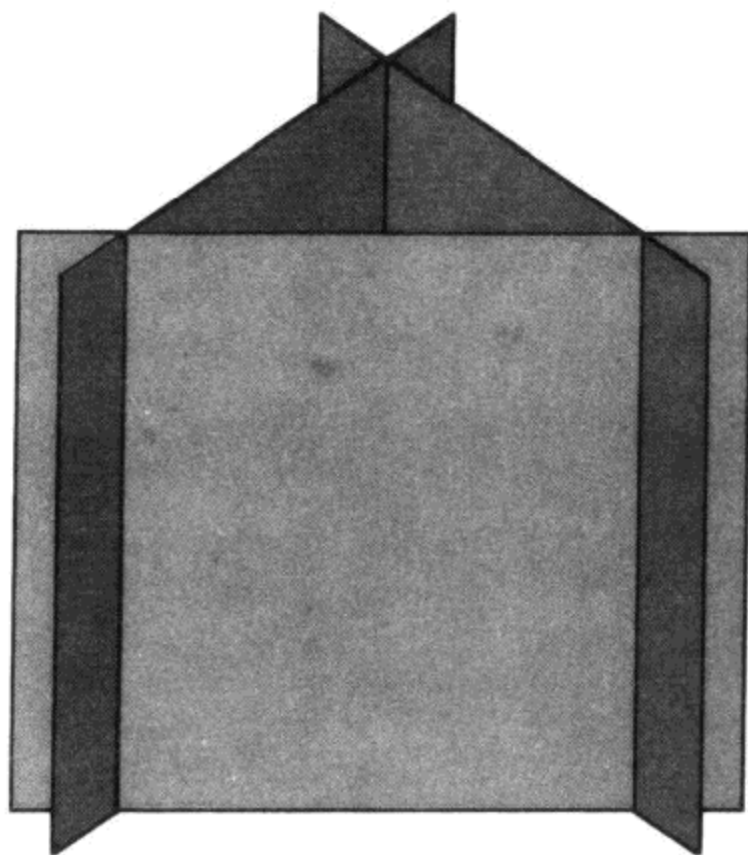


图 4.6 三个平面不一定相交于一点

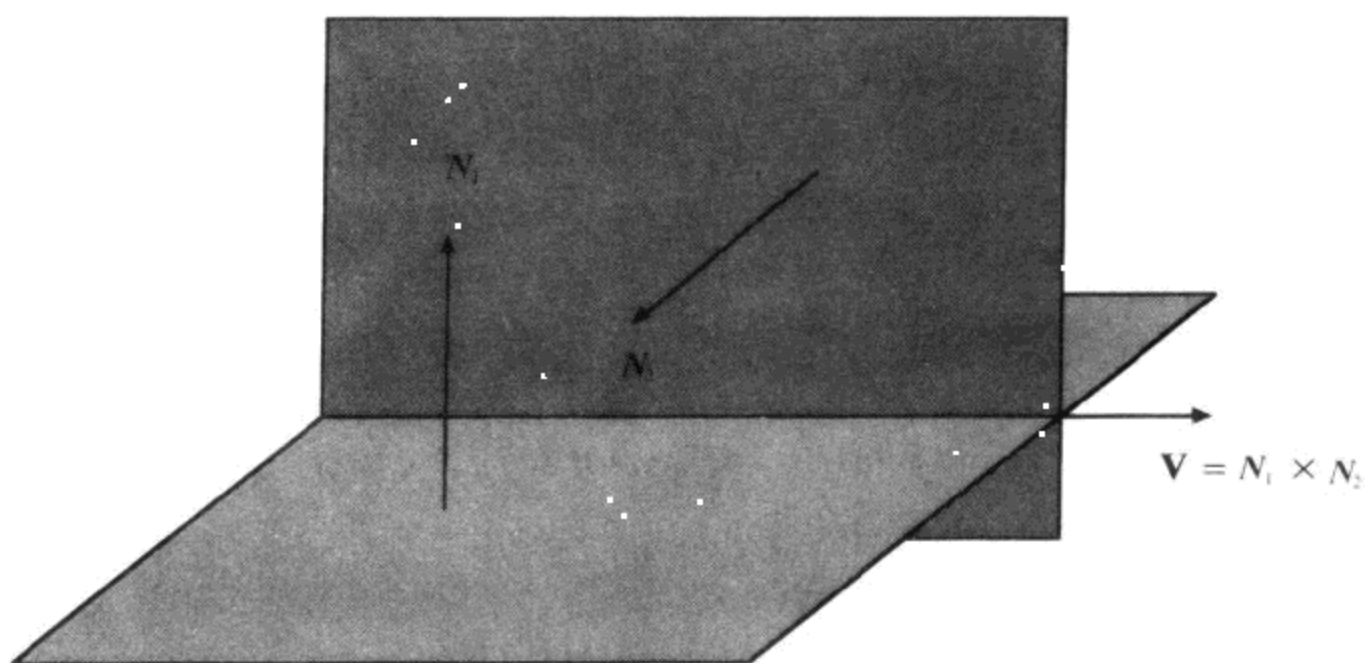


图 4.7 两个不平行的平面相交于一条与两个平面的法向量都垂直的直线

在有了方向 \mathbf{V} 的情况下,为了完整地表示一条直线,需要给出位于该直线上的一个点。这可以通过构造过原点且法向量为 \mathbf{V} 的第三个平面来实现。在这种情况下,肯定可以解出三个平面的交点,如图 4.8 所示。根据等式(4.20),可以计算出位于交线上的点 Q

$$Q = \begin{bmatrix} (N_1)_x & (N_1)_y & (N_1)_z \\ (N_2)_x & (N_2)_y & (N_2)_z \\ V_x & V_y & V_z \end{bmatrix}^{-1} \begin{bmatrix} -D_1 \\ -D_2 \\ 0 \end{bmatrix} \quad (4.21)$$

平面 L_1 和平面 L_2 的交线可以表示为 $P(t) = Q + t\mathbf{V}$ 。

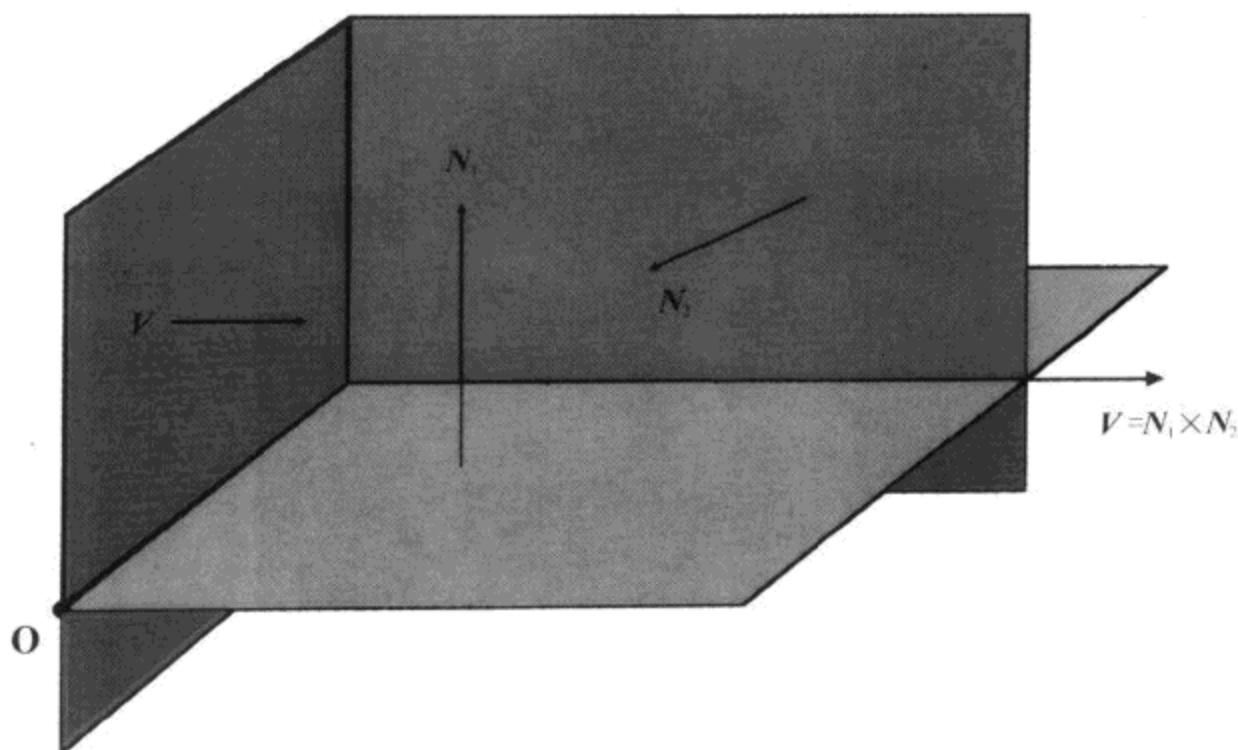


图 4.8 构造第三个平面(过原点且法向为两平面的交直线所指的方向)

4.2.3 平面的变换

这里我们要使用 3×3 矩阵 M 和 3D 平移向量 T 对平面进行变换。前面讲过,可以使用 M 的逆转置矩阵对法向量 N 进行变换,除此之外,在平面的变换过程中还要考虑平面距原点的带符号距离。如果知道了位于过原点的平面上的一点 P_0 ,那么就能使用下面的公式计算出从原点到变换平面的带符号距离 D'

$$\begin{aligned} D' &= -((M^{-1})^T N) \cdot (MP_0 + T) \\ &= -((M^{-1})^T N)^T MP_0 - ((M^{-1})^T N)^T T \\ &= -N^T M^{-1} MP_0 - N^T M^{-1} T = D - N \cdot M^{-1} T \end{aligned} \quad (4.22)$$

这里回顾一下等式(3.26), 4×4 矩阵 F 的逆矩阵可由一个 3×3 矩阵 M 和一个 3D 平移向量 T 表示

$$F^{-1} = \left[\begin{array}{c|c} M^{-1} & -M^{-1}T \\ \hline 0 & 1 \end{array} \right] \quad (4.23)$$

可以进一步得到 F^{-1} 的转置:

$$(F^{-1})^T = \left[\begin{array}{c|c} (M^{-1})^T & 0 \\ \hline -M^{-1}T & 1 \end{array} \right] \quad (4.24)$$

$D - N \cdot M^{-1} T$ 的值恰好等于 $(F^{-1})^T$ 的第四行和 4D 向量 $\langle N_x, N_y, N_z, D \rangle$ 的点积。这表明可以把平面看成四维向量,除了需要用到 4×4 变换矩阵的逆转置矩阵外,可以用与三维法向量同样的变换方法对平面进行变换。所以,使用 4×4 矩阵 F ,平面 $L = \langle N, D \rangle$ 变换成

$$L' = (F^{-1})^T L \quad (4.25)$$

4.3 视截体

图4.9给出了视截体的示意图,视截体用来表示一个空间范围,位于这个空间范围内的三维场景的任何物体都可以被看到。视截体的形状是一个棱锥,其顶点表示用来观看场景的摄像机的位置,之所以具有这样的形状,是因为要用视截体来表示摄像机透过一个矩形窗口(实际上就是计算机屏幕)所能够见到的实际范围。视截体由六个平面围成,其中的四个平面与场景的边界相对应,分别被称为左、右、底、顶视截面。另外两个平面称为近视截面和远视截面,它们定义了最近和最远距离,对于场景中的物体,只有位于该距离范围内的物体才能够被摄像机所看到。

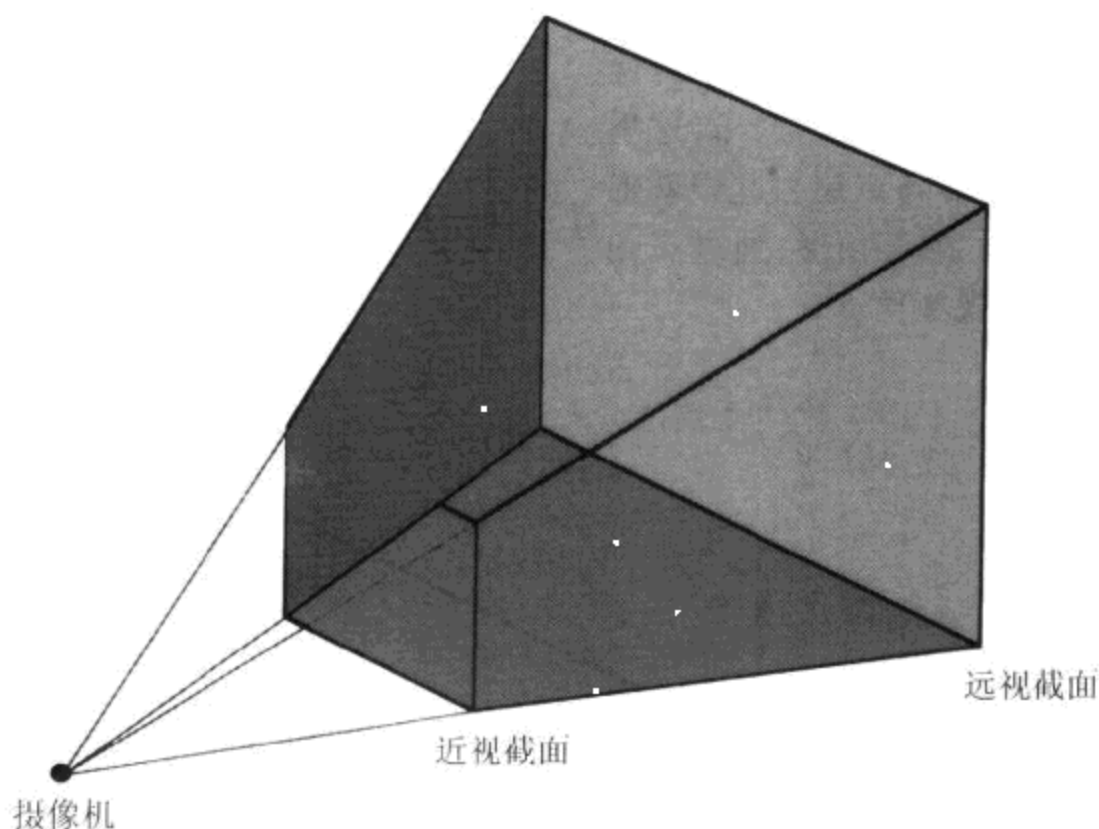


图4.9 视截体

视截体与摄像机空间是一致的。摄像机空间,也称眼球空间,它是一个坐标系,在该坐标系中,摄像机位于原点, x 轴指向右边, y 轴指向上方, z 轴的方向取决于所用的3D图形库。在OpenGL环境下, z 轴方向是摄像机所指方向的反方向,形成了右手坐标系,如图4.10所示。在Direct3D环境下, z 轴方向与摄像机所指的方向一致,形成了左手坐标系。

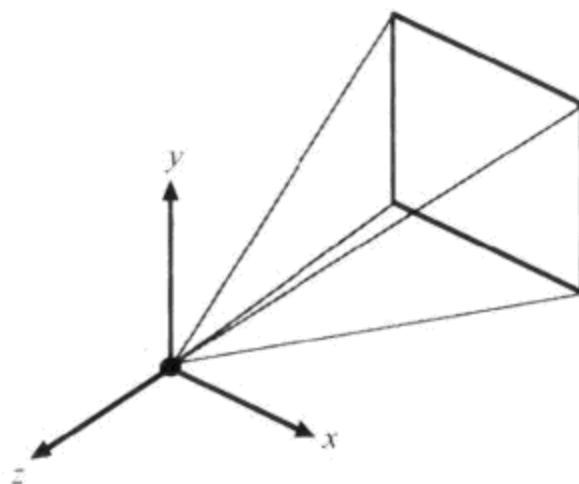


图4.10 OpenGL的摄像机空间

4.3.1 视域

如图4.11所示,投影面是垂直于摄像机视线方向的一个平面,该平面位于与摄像机的距离为 e 的位置,分别与左右视截面交于 $x = -1$ 和 $x = 1$ 处。距离 e 有时称为摄像机的焦距,它和左右两个视截面间的夹角 α 有关。夹角 α 称为水平视域角。

如果知道了水平视域角 α 的值,就可以根据三角关系计算出投影面的距离 e

$$e = \frac{1}{\tan(\alpha/2)} \quad (4.26)$$

焦距越短,则视域越宽。通过逐渐缩小视域角,可以放大摄像机所能看到的物体,同时焦距也在变长。

显示器的屏幕高宽比等于它的高度除以宽度得到的值,比如一个 640×480 像素的显示器的屏幕高宽比为 0.75。由于大多数显示器不是正方形而是长方形的,所以垂直视域与水平视域是不相等的。底视截面和顶视截面与投影面相交于 $y = \pm a$,这里的 a 就是显示器的屏幕高宽比。这样就形成了如图 4.12 中所示的三角形,通过三角关系可以计算出垂直视域角 β

$$\beta = 2 \tan^{-1}(a/e) \quad (4.27)$$

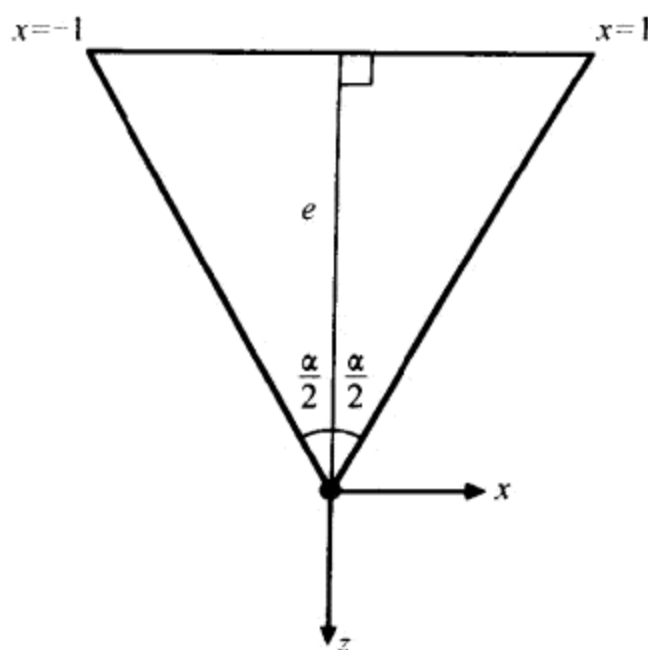


图 4.11 摄像机到投影面的距离取决于水平视域角

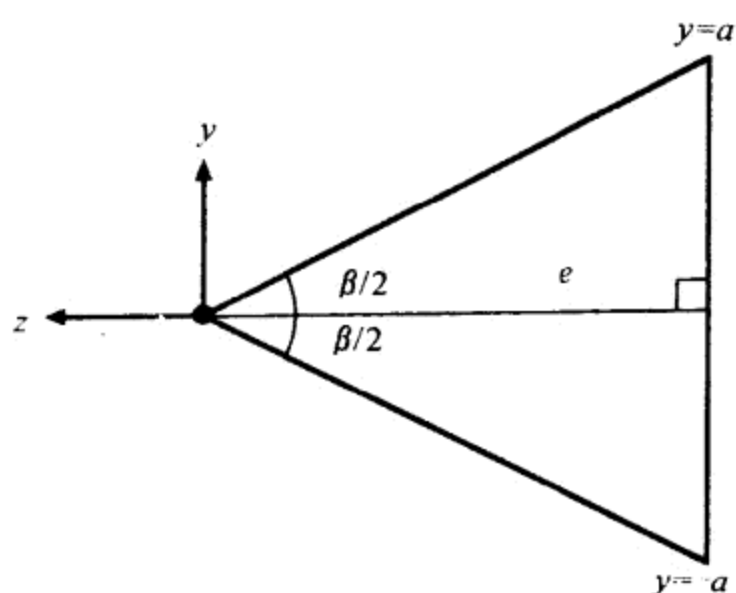


图 4.12 垂直视域角取决于屏幕高宽比

视截体的四个边平面在投影面上切出一个与摄像机的距离为 e ,且边为 $x = \pm 1$ 和 $y = \pm a$ 的矩形。OpenGL 中的函数 `glFrustum()` 要求在距离摄像机 n (n 就是近视截面的距离) 处给定一个矩形。可以用因子 n/e 对该矩形进行缩放,这样左边界位于 $x = -an/e$,右边界位于 $x = n/e$,底边界位于 $y = -an/e$,顶边界位于 $y = an/e$ 。

4.3.2 视截面

在图 4.13 中给出了表示摄像机空间的六个视截面的法向量方向。四个边平面的内法向量可通过将四条边的方向向量向视截体中心旋转 90° 而得到。四个边平面都通过原点,所以都有 $D = 0$ 。近视截面在它的法向上与原点的距离为 n ,因此有 $D = -n$ 。远视截

面在它的法向与原点的距离为 f , 故 $D = f$ 。

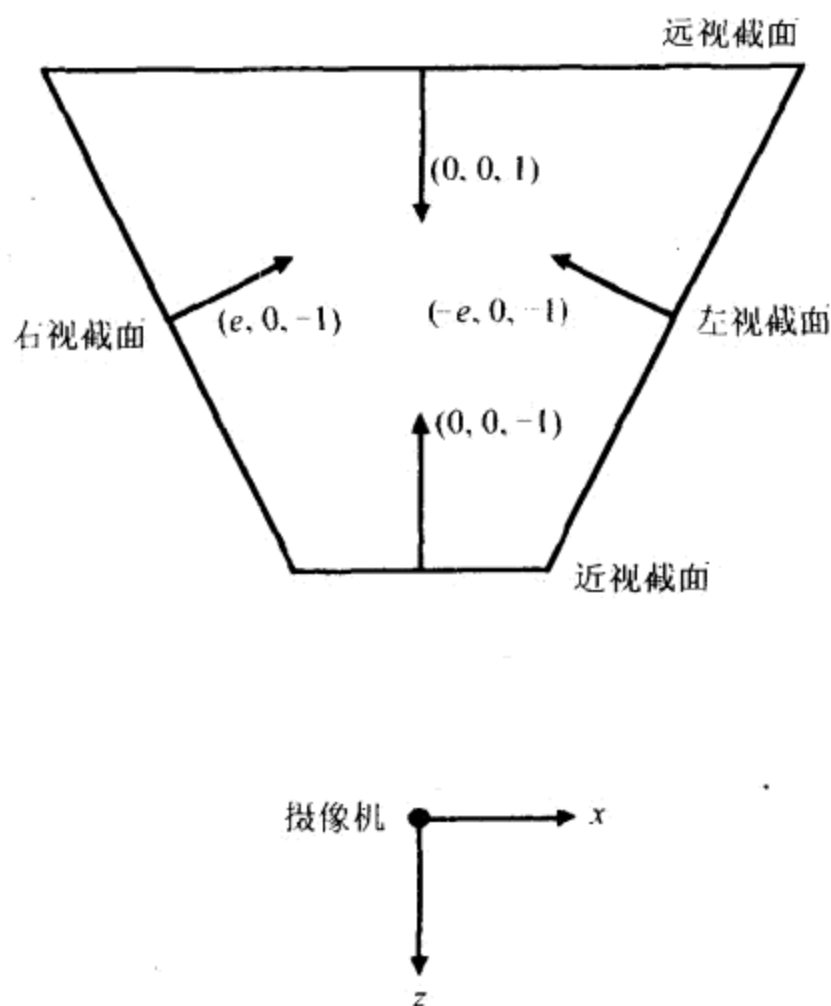


图 4.13 OpenGL 的摄像机空间所看到的视截面的法线方向

表 4.1 归纳了与视载体的六个边平面相对应的四维平面向量。在表中, 四个边平面的法向量已经被规格化为单位长度。

表 4.1 OpenGL 的摄像机空间所看到的平面向量

平面	$\langle N, D \rangle$
近	$\langle 0, 0, -1, -n \rangle$
远	$\langle 0, 0, 1, f \rangle$
左	$\left\langle \frac{e}{\sqrt{e^2 + 1}}, 0, -\frac{1}{\sqrt{e^2 + 1}}, 0 \right\rangle$
右	$\left\langle -\frac{e}{\sqrt{e^2 + 1}}, 0, -\frac{1}{\sqrt{e^2 + 1}}, 0 \right\rangle$
上	$\left\langle 0, -\frac{e}{\sqrt{e^2 + a^2}}, -\frac{a}{\sqrt{e^2 + a^2}}, 0 \right\rangle$
下	$\left\langle 0, \frac{e}{\sqrt{e^2 + a^2}}, -\frac{a}{\sqrt{e^2 + a^2}}, 0 \right\rangle$

注: 其中 e 为焦距, a 为屏幕高宽比, 近视截面的距离为 n , 远视截面的距离为 f 。

4.4 透视校正插值

当 3D 图形处理器将一个三角形渲染到屏幕上的时候,需要在屏幕上对三角形以逐行扫描的方式进行光栅化。三角形的顶点除了携带在摄像机空间中的位置信息外,还携带有其他信息,比如颜色、亮度及纹理映射坐标,在光栅化的过程中必须在三角形的表面上对这些信息进行插值。当画出三角形的一条扫描线时,扫描线上的每个像素的信息,是对扫描线的左右端点处已知的信息值进行插值运算而得到的。

如图 4.14 所示,对三角形表面的校正插值是非线性的,这是因为对于投影面上相等的空间步长,它们在三角形面上对应的步长会随着离摄像机的距离的增加而变长。图形处理器必须对纹理映射坐标采用非线性的方法进行插值,否则会出现纹理映射失真。对于顶点所携带的其他信息(如光照的颜色),现在的硬件都会对其进行非线性插值,而老的图形卡只是对它们进行简单的线性插值,因为对这些信息的非线性插值和线性插值所造成的区别,不像纹理映射时那样明显。

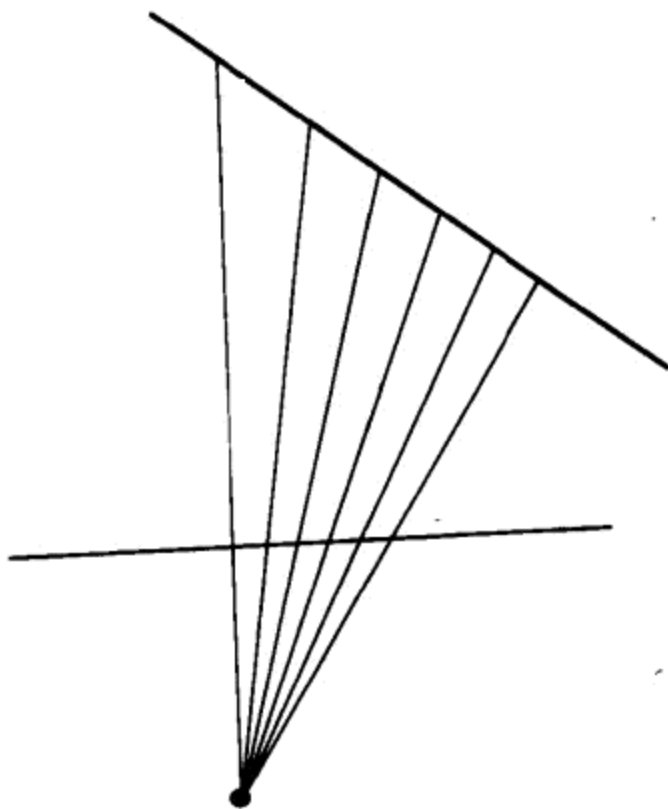


图 4.14 在三角形表面上的校正插值是非线性的

4.4.1 深度插值(depth interpolation)

这里需要注意的是,3D 图形硬件对三角形面上的点的 z 坐标(表示深度)进行插值运算时用的是线性插值法,这与本节所讨论的透视校正法不同,在后面的 4.5.1 节中将对透视投影矩阵进行讨论,到时会对这一点进行解释。

在图 4.15 中给出了位于 xz 平面上的一条线段,它对应于三角形的一条扫描线。在光栅化的过程中要对该线段上的点进行采样,采样时先在投影平面上取空间等距点(这里的空间等距点对应于屏幕的像素点),然后求通过这些等距点的光线与线段的交点,得到的所有交点就是采样点。如果线段所在的直线不通过原点(否则三角形是边界可见的,三角形本身

不可见),则可以用下面的方程来描述这条直线

$$ax + bz = c \tag{4.28}$$

其中 $c \neq 0$ 。

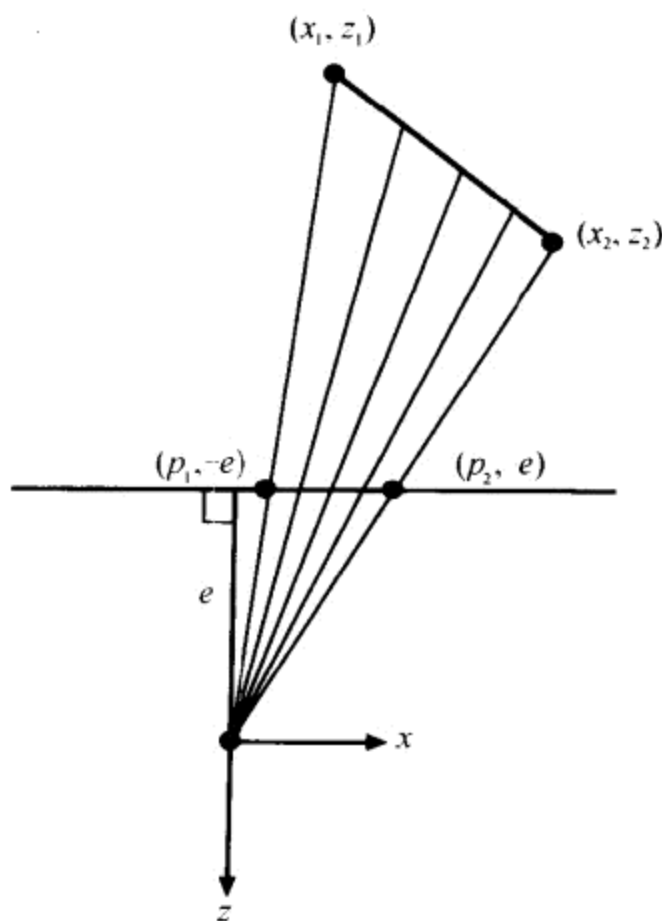


图 4.15 三角形的扫描线所对应的线段的采样

对于直线上的一点 (x, z) , 可以引一条从原点(摄像机位置)指向点 (x, z) 的射线, 并求得射线与投影平面的交点。在投影平面上的 z 坐标恒为 $-e$ 。可以从图 4.15 给出的相似三角形关系中导出下面的关系式, 从而求得点 (x, z) 在投影平面上所对应 x 的坐标 p

$$\frac{-p}{x} = \frac{-e}{z} \tag{4.29}$$

解关于 x 的方程, 并将 x 代回等式(4.28), 可以将直线的方程重写为如下形式:

$$\left(\frac{-ap}{e} + b\right)z = c \tag{4.30}$$

将上式改写为只有一边出现 $1/z$ 的形式, 可以得到便于以后使用的方程:

$$\frac{1}{z} = -\frac{ap}{ce} + \frac{b}{c} \tag{4.31}$$

设线段的两个端点为 (x_1, z_1) 和 (x_2, z_2) , 它们在投影平面上的投影分别为 $(p_1, -e)$ 和 $(p_2, -e)$, 同时设 $p_3 = (1-t)p_1 + tp_2 (0 \leq t \leq 1)$ 是投影平面上插值点的 x 坐标, 这里需求出射线穿过点 $(p_3, -e)$ 与三角形面的交点 (x_3, z_3) 的 z 坐标。将 $p_3 = (1-t)p_1 + tp_2$ 和 z_3 代到等式(4.31), 可以得到

$$\begin{aligned} \frac{1}{z_3} &= -\frac{ap_3}{ce} + \frac{b}{c} \\ &= -\frac{ap_1}{ce}(1-t) - \frac{ap_2}{ce}t + \frac{b}{c} \end{aligned}$$

$$\begin{aligned}
 &= \left(-\frac{ap_1}{ce} + \frac{b}{c}\right)(1-t) + \left(-\frac{ap_2}{ce} + \frac{b}{c}\right)t \\
 &= \frac{1}{z_1}(1-t) + \frac{1}{z_2}t
 \end{aligned} \tag{4.32}$$

这个结果表明,在整个三角面上, z 坐标的倒数恰好是按线性的方式进行插值的。

4.4.2 顶点属性的插值

从现在开始,顶点所携带的信息,例如光照颜色和纹理映射坐标,统一称之为顶点属性。当对三角形的面进行光栅化时,必须在整个三角面上对三角形顶点的属性进行插值。这里假设扫描线的端点的深度坐标值为 z_1 和 z_2 ,且各自拥有的标量属性为 b_1 和 b_2 。我们希望属性插值 b_3 会满足这样的条件: b_3 与线段属性的总差值的比例,等于深度插值 z_3 与线段 z 值的总差值的比例,即满足下面的等式:

$$\frac{b_3 - b_1}{b_2 - b_1} = \frac{z_3 - z_1}{z_2 - z_1} \tag{4.33}$$

由等式(4.32)可以得到

$$z_3 = \frac{1}{\frac{1}{z_1}(1-t) + \frac{1}{z_2}t} \tag{4.34}$$

将该值代入等式(4.33)可以解出 b_3

$$b_3 = \frac{b_1 z_2 (1-t) + b_2 z_1 t}{z_2 (1-t) + z_1 t} \tag{4.35}$$

分子分母同乘以 $\frac{1}{z_1 z_2}$,则可以从等式右边取出因子 z_3 ,得

$$b_3 = \frac{\frac{b_1}{z_1}(1-t) + \frac{b_2}{z_2}t}{\frac{1}{z_1}(1-t) + \frac{1}{z_2}t} \tag{4.36}$$

这表明,在整个三角面上可以用线性的方式对 b/z 进行插值。当对扫描线进行光栅化时,图形处理器首先计算得到 $1/z$ 的线性插值,然后取其倒数,并乘以 b/z 的线性插入值,这样就可以得到顶点任意属性 b 的透视校正插入值。

4.5 投 影

在二维的显示器屏幕上显示三维场景时,需要确定三维场景中的每个顶点在屏幕上的对应的显示位置。前面已经讲过,对于位置为 P 的顶点,可以引一条从原点出发并经过 P 的射线,求射线与投影平面的交点,就可以确定顶点在投影平面上的投影位置。投影点的 x 和 y 坐标由下面公式给出,即

$$\begin{aligned}
 x &= -\frac{e}{p_z} p_x \\
 y &= -\frac{e}{p_z} p_y
 \end{aligned} \tag{4.37}$$

(这里需要注意, p_z 的值是负的, 因为摄像机位于 z 轴的负方向上)

利用上面的公式计算 z 坐标时, 总是得到一个值为 $-e$ 的投影深度。但是, 在消除隐藏的时候, 需要更有用的深度信息, 因此在 3D 图形系统中要改用相似坐标, 在四维空间中对顶点进行投影。

4.5.1 透视投影

透视投影的作用是将 x 和 y 坐标映射到投影平面的正确的位置上, 同时保持深度信息不变。透视投影是通过一个将视截体映射到一个立方体中的过程来完成的, 如图 4.16 所示, 这个立方体称为齐次裁剪空间, 在 OpenGL 中它以原点为中心, 在 x 、 y 和 z 轴上的取值范围都是从 -1 到 $+1$ 。

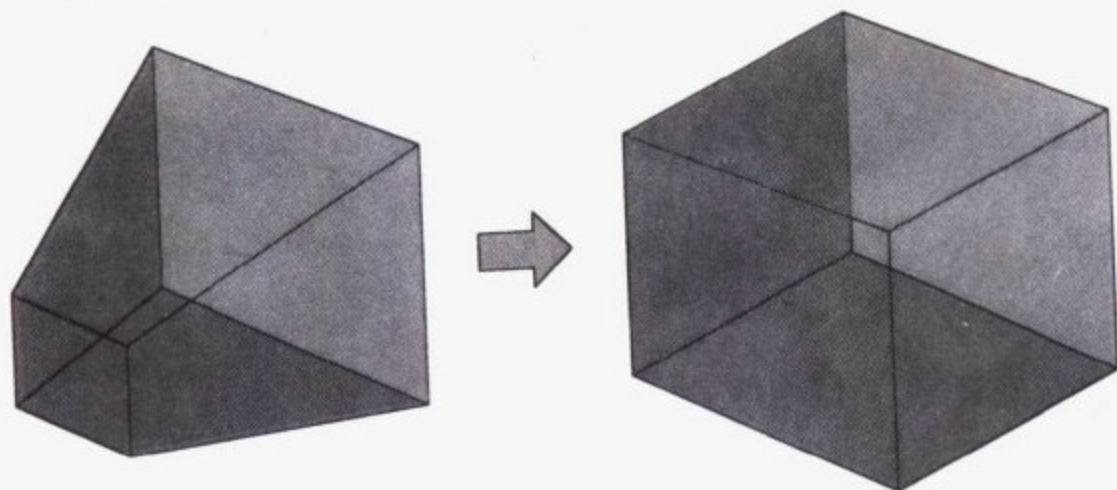


图 4.16 透视投影将视截体映射到立方体(齐次裁剪空间)上

到齐次裁剪空间的映射是通过一个 4×4 投影矩阵实现的。在这个投影矩阵中, 除了其他的变换功能外, 还要保证变换后的点的 w 坐标等于摄像机空间中的点的 z 坐标的负值, 这样, 将变换后点的坐标分量除以 w 坐标后, 就可以生成裁剪空间中相应的三维点。

设 $\mathbf{P} = \langle P_x, P_y, P_z, 1 \rangle$ 是摄像机空间中的齐次坐标点, 并且位于裁剪视截体之内。在 OpenGL 中, 设置视截体的函数 `glFrustum()` 需要一些参数, 其中包括视截体的四个边平面在近视截面上所截出的矩形区域的左边 $x=l$ 、右边 $x=r$ 、底边 $y=b$ 和顶边 $y=t$ 。假设近视截面位于 $z=-n$, 可以通过下面的公式来计算点 \mathbf{P} 在近视截面上的投影的 x 坐标和 y 坐标, 即

$$x = -\frac{n}{P_z} P_x \quad (4.38)$$

$$y = -\frac{n}{P_z} P_y$$

视截体里的任何点投影到近视截面上都会满足 $l \leq x \leq r$ 和 $b \leq y \leq t$, 这里要将这些区间映射到区间 $[-1, 1]$ 上, 以满足视截体到齐次裁剪空间的映射需要。这可以通过简单的线性函数实现

$$x' = (x - l) \frac{2}{r - l} - 1 \quad (4.39)$$

$$y' = (y - b) \frac{2}{t - b} - 1 \quad (4.40)$$

将公式(4.38)代入上面的公式,化简后可得

$$x' = \frac{2n}{r-l} \left(-\frac{P_x}{P_z} \right) - \frac{r+l}{r-l} \quad (4.41)$$

$$y' = \frac{2n}{t-b} \left(-\frac{P_y}{P_z} \right) - \frac{t+b}{t-b} \quad (4.42)$$

将投影 z 坐标映射到区间 $[-1, 1]$, 需要进行比较复杂的计算。由于点 P 位于视截体之内, 因此它的 z 坐标值满足 $-f \leq p_z \leq -n$, 其中 n, f 分别是摄像机点到近视截面和远视截面的距离。我们希望找到一个将 $-n \rightarrow -1$ 映射到 $-f \rightarrow 1$ 的函数(因为这样的映射是对 z 轴的反射, 可以知道齐次裁剪空间是左手向的)。在扫描光栅化过程中, 必须对 z 坐标的倒数进行插值, 因此这里需要建立关于 $1/z$ 的映射函数, 这样就可以对投影深度值进行线性插值。对应的映射函数形式为

$$z' = \frac{A}{z} + B \quad (4.43)$$

将已知的映射 $-n \rightarrow -1$ 和 $-f \rightarrow 1$ 代入上式, 可以求得未知的系数 A 和 B

$$-1 = \frac{A}{-n} + B \quad (4.44)$$

$$1 = \frac{A}{-f} + B$$

经过简单的代数计算, 可以求得 A, B 的值为

$$A = \frac{2nf}{f-n} \quad (4.45)$$

$$B = \frac{f+n}{f-n}$$

这样, z 坐标就可以通过下面的函数映射到区间 $[-1, 1]$, 即

$$z' = -\frac{2nf}{f-n} \left(-\frac{1}{P_z} \right) + \frac{f+n}{f-n} \quad (4.46)$$

在公式(4.41)、(4.42)和(4.46)都包含有对 $-P_z$ 的除法。3D 点 $\tilde{P}' = \langle x', y', z' \rangle$ 等价于除以 z 坐标之后的 4D 齐次坐标点 $P' = \langle -x'P_z, -y'P_z, -z'P_z, -P_z \rangle$ 。由于 $-x'P_z$, $-y'P_z$ 和 $-z'P_z$ 的值由下面的公式给出:

$$-x'P_z = \frac{2n}{r-l}P_x + \frac{r+l}{r-l}P_z \quad (4.47)$$

$$-y'P_z = \frac{2n}{t-b}P_y + \frac{t+b}{t-b}P_z \quad (4.48)$$

$$-z'P_z = -\frac{f+n}{f-n}P_z - \frac{2nf}{f-n} \quad (4.49)$$

它们都是关于 P 点坐标的线性函数, 所以可以用下面的 4×4 矩阵以统一的方式计算点 P' 。

$$P' = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2nf}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix} P \quad (4.50)$$

公式(4.50)中的矩阵就是在 OpenGL 中由函数 `glFrustum()` 生成的透视投影矩阵。这个矩阵将摄像机空间中的点变换成裁剪坐标空间中的齐次坐标点,并且使得到的齐次坐标点的 w 坐标等于原摄像机空间 z 坐标的负值。对顶点的属性进行插值时(参见 4.4.2 节),实际上就是将这个 w 坐标的倒数作为公式(4.36)中的 z 值进行插值运算。

4.5.2 正投影

正投影也称为平行投影,在这种投影中不会出现透视失真的现象(比如远小近大现象)。如图 4.17 所示,对于摄像机空间中的点,可以引出平行于摄像机视线方向的射线,点沿着射线的方向映射到投影平面。

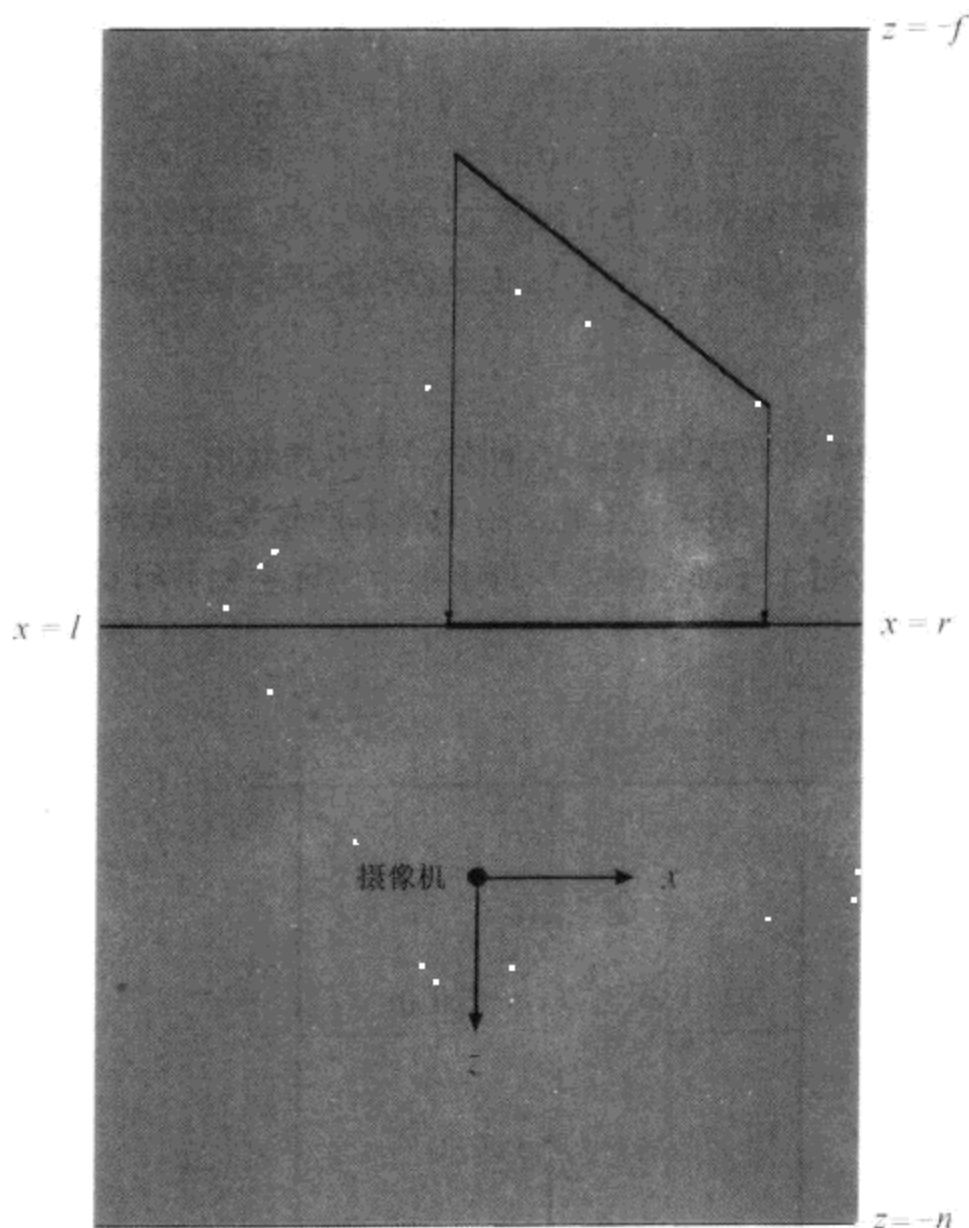


图 4.17 正射投影

正投影的视野范围是由 xy 平面上的一个矩形和近视截面与远视截面的距离来确定的。由于不存在透视失真现象,在正投影中可以对三角形的深度值进行线性插值,所以到齐次裁剪空间的映射可以在三个轴上均以线性的方式实现。将 x 坐标和 y 坐标从区间 $[l, r]$ 和 $[b, t]$ 映射到区间 $[-1, 1]$ 上的函数为

$$x' = \frac{2}{r-l}x - \frac{r+l}{r-l} \quad (4.51)$$

$$y' = \frac{2}{t-b}y - \frac{t+b}{t-b} \quad (4.52)$$

z 轴方向的映射与此类似,只不过要将 z 值取负($-n \rightarrow -1, -f \rightarrow 1$),可以用下面的函数将 z 坐标从区间 $[-f, -n]$ 映射到区间 $[-1, 1]$,即

$$z' = \frac{-2}{f-n}z - \frac{f+n}{f-n} \quad (4.53)$$

可以将上面三个函数合写成矩阵形式为

$$P' = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & \frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & \frac{t+b}{t-b} \\ 0 & 0 & \frac{-2}{f-n} & \frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix} P \quad (4.54)$$

式(4.54)中的矩阵就是 OpenGL 中由函数 `glOrtho()` 生成的正投影矩阵。需要注意的是,经过正投影变换后的 w 坐标仍然为 1,这正是没有出现透视投影的原因。

4.5.3 提取视截面

对于任一投影矩阵 M ,都可以用很简单的方法从中提取出一个四维向量,这个四维向量与摄像机空间的六个视截面对应。这里介绍的技术的依据是裁剪空间的平面总是相同的。如图 4.18 所示,由于每个平面法线都平行于某个坐标主轴,所以得到它们很容易。

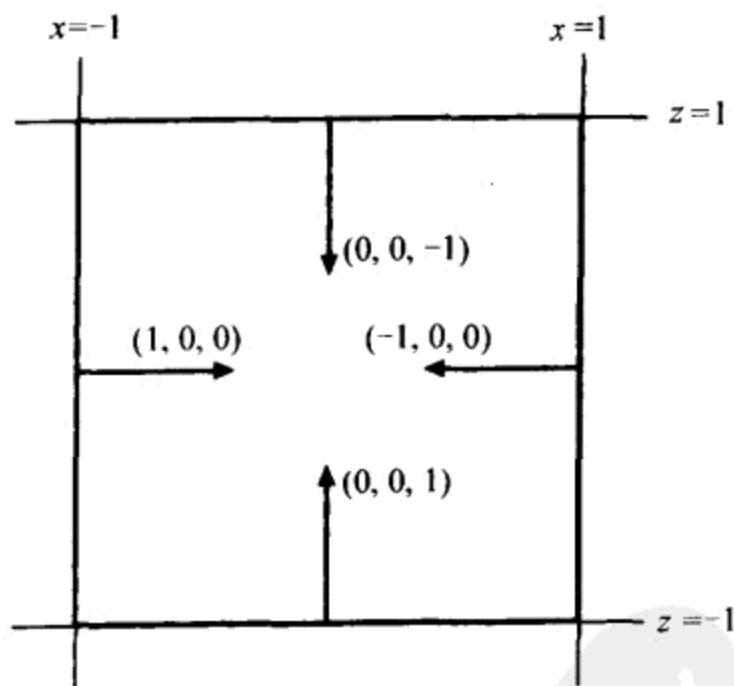


图 4.18 围成立方体齐次裁剪空间的四维平面向量

假设 L' 是围成裁剪空间的六个平面中的一个。矩阵 M 的逆矩阵实现了从裁剪空间到摄像机空间的变换,由于所有的平面都是使用逆转置矩阵进行变换的,所以对应于裁剪空间平面 L' 对应的摄像机空间平面 L 可由下式表示为

$$L = ((M^{-1})^{-1})^T L' = M^T L' \quad (4.55)$$

在表 4.2 中列出了裁剪空间的六个平面向量。由于每个平面向量都包含两个非零项,并且非零项都为 ± 1 ,所以可以将摄像机空间的每个视截面写成矩阵 M^T 的两列的和或差,这也等价于矩阵 M 的两行的和或差。

表 4.2 剪切空间平面向量

平面	$\langle N, D \rangle$
近	$\langle 0, 0, 1, 1 \rangle$
远	$\langle 0, 0, -1, 1 \rangle$
左	$\langle 1, 0, 0, 1 \rangle$
右	$\langle -1, 0, 0, 1 \rangle$
上	$\langle 0, 1, 0, 1 \rangle$
下	$\langle 0, -1, 0, 1 \rangle$

如果使用符号 M_i 表示矩阵 M 的第 i 行,则可以用下面的公式来表示摄像机空间的视截面,因为由这些公式产生的平面向量不具有单位长度的平面法向量,所以需要对他们进行适当的缩放调整。

$$\begin{aligned}
 \text{near} &= M_4 + M_3 \\
 \text{far} &= M_4 - M_3 \\
 \text{left} &= M_4 + M_1 \\
 \text{right} &= M_4 - M_1 \\
 \text{bottom} &= M_4 + M_2 \\
 \text{top} &= M_4 - M_2
 \end{aligned} \tag{4.56}$$

这些公式适用于任何投影矩阵。这里需要强调的是,在计算视截面时,如果已知某一视截体的焦距和屏幕高宽比,那么使用表 4.1 中给出的公式会更加有效。

4.6 本章小结

直线

通过点 P_0 且方向平行于方向 V 的直线可以表示为

$$P(t) = P_0 + tV$$

点 Q 到直线 $P(t)$ 的距离为

$$d = \sqrt{(Q - P_0)^2 - \frac{[(Q - P_0) \cdot V]^2}{V^2}}$$

平面

法线方向为 N 且包含点 P_0 的平面可表示为

$$N \cdot P + D = 0$$

其中 $D = -\mathbf{N} \cdot \mathbf{P}_0$ 。这也可以表示为 $\mathbf{L} \cdot \mathbf{P} = 0$, 其中 \mathbf{L} 是 4D 向量 $\langle \mathbf{N}, D \rangle$, \mathbf{P} 是 w 坐标为 1 的齐次坐标点。从点 Q 到平面 L 的距离可以简单地表示为 $\mathbf{L} \cdot \mathbf{Q}$ 。

将变换点的变换矩阵的逆转置矩阵用于平面的变换。

直线和平面求交

直线 $\mathbf{P}(t) = \mathbf{Q} + t\mathbf{V}$ 和平面 L 的交点 t 为

$$t = -\frac{\mathbf{L} \cdot \mathbf{Q}}{\mathbf{L} \cdot \mathbf{V}}$$

视截体

水平视域角为 α 的视截体的焦距 e 为

$$e = \frac{1}{\tan(\alpha/2)}$$

对于屏幕高宽比为 a 的显示器来说, 与摄像机距离为 n 的近视截面上被裁出一个矩形, 该矩形的边界为 $x = \pm n/e$ 和 $y = \pm an/e$ 。

透视校正插补

在透视投影中, 深度插值是通过深度值 z_1 和 z_2 倒数的线性插值而得到的

$$\frac{1}{z_3} = \frac{1}{z_1}(1-t) + \frac{1}{z_2}t$$

顶点的属性的透视校正插值使用类似的公式

$$\frac{b_3}{z_3} = \left[\frac{b_1}{z_1}(1-t) + \frac{b_2}{z_2}t \right]$$

其中 b_1 和 b_2 是顶点属性值。

透视投影

将点从摄像机空间变换到裁剪空间的透视投影矩阵为

$$\begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2nf}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

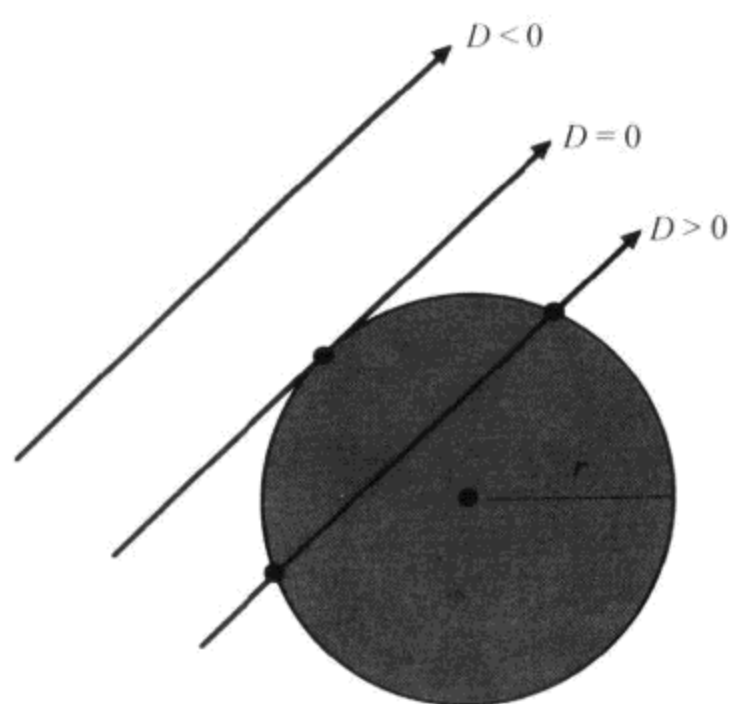
其中 n 和 f 分别是近视截面和远视截面与摄像机距离, l, r, b 和 t 是在近视截面上裁剪出的矩形视图的左、右、下、上边。

习 题

1. 计算四维向量 $\langle N, D \rangle$, 该向量表示通过 $\langle 1, 2, 0 \rangle$, $\langle 2, 0, -1 \rangle$ 和 $\langle 3, -2, 1 \rangle$ 三个点的平面。
2. 设某一视截体的水平视域角为 75° , 计算一个分辨率为 1280×1024 像素的显示器的对应垂直视域角。
3. 设视截体的水平视域角是 90° , 屏幕高宽比为 0.75 , 试计算其左、右、上、下视截面。
4. 在齐次裁剪空间中, 如果坐标的区间是 $[0, 1]$ 而不是 $[-1, 1]$, 按照推导公式(4.50)的方法, 推导出对 $-n \rightarrow 0$ 和 $-f \rightarrow 1$ 进行映射的透视投影矩阵。



第 5 章 光线跟踪



术语“光线跟踪”指的是跟随光束以求得与光线相互作用的物体的算法。光线跟踪的应用包括光谱的产生、可见性确定、碰撞检测以及视线检测。本章主要介绍当光线照射到物体上时,如何确定光线与物体的交点,以及当光线照射到反射面或折射面时,光线的传播路径是如何改变的。

5.1 根的求解

求解由方程

$$P(t) = Q + tV \quad (5.1)$$

所确定的直线与一个曲面的交点的问题时,通常要求求解关于 t 的 n 次多项式的根。对于平面,多项式的次数为 1,求解过程很简单。对于二次曲面,比如球面或圆柱面,多项式的次数为 2,通过求解二次方程就可得到所需的根。至于更复杂的曲面,比如样条曲面和环形圆纹曲面,所对应的多项式的次数为 3 或 4,在这种情况下仍然能够通过解析的方法来求解,但需要花费更大的计算开销。

关于二次、三次和四次多项式的解析求解法会在本章中给予介绍,但关于三次方程和四次方程求解的完整推导过程,由于超出了本书的范围,不予更多介绍。另外,在本章中还会对 Newton-Raphson 迭代法的求根方法进行介绍。

5.1.1 二次多项式

对关于 t 的二次多项式的根,通过简单的代数运算解下面的方程就可以求得:

$$at^2 + bt + c = 0 \quad (5.2)$$

在方程式的两边同时减去 c ,再同时除以 a 可得

$$t^2 + \frac{b}{a}t = -\frac{c}{a} \quad (5.3)$$

在方程的两边同时加上 $b^2/4a^2$ 来配方,结果如下:

$$t^2 + \frac{b}{a}t + \frac{b^2}{4a^2} = -\frac{c}{a} + \frac{b^2}{4a^2} \quad (5.4)$$

将方程的左边写成平方的形式,同时将方程的右边写成同分母的形式:

$$\left(t + \frac{b}{2a}\right)^2 = \frac{b^2 - 4ac}{4a^2} \quad (5.5)$$

方程的两边开平方根,然后都加上 $b/2a$,可以得到

$$t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (5.6)$$

这就是著名的二次求根公式。 $D = b^2 - 4ac$ 称为多项式的判别式,它确定了多项式实根的个数。如果 $D > 0$,则存在两个实根;如果 $D = 0$,则存在一个实根;如果 $D < 0$,则不存在实根。通过计算判别式的值,在不需计算实际交点的情况下就可以知道光线与物体是否相交。

5.1.2 三次多项式

三次方程具有如下的形式:

$$t^3 + at^2 + bt + c = 0 \quad (5.7)$$

这里已经做了必要的除法,以使最高次项的系数为1。做如下的替换:

$$t = x - \frac{a}{3} \quad (5.8)$$

这样可以去掉多项式的二次项,于是得到下面的方程:

$$x^3 + px + q = 0 \quad (5.9)$$

其中

$$\begin{aligned} p &= -\frac{1}{3}a^2 + b \\ q &= \frac{2}{27}a^3 - \frac{1}{3}ab + c \end{aligned} \quad (5.10)$$

只要求得方程(5.9)的解 x ,将 x 减去 $a/3$ 就可得到方程(5.7)的解 t 。

三次多项式的判别式为

$$D = -4p^3 - 27q^2 \quad (5.11)$$

令

$$\begin{aligned} r &= \sqrt[3]{-\frac{1}{2}q + \sqrt{-\frac{1}{108}D}} \\ s &= \sqrt[3]{-\frac{1}{2}q - \sqrt{-\frac{1}{108}D}} \end{aligned} \quad (5.12)$$

则可以将方程(5.9)的三个复数根表示为

$$\begin{aligned} x_1 &= r + s \\ x_2 &= \rho r + \rho^2 s \\ x_3 &= \rho^2 r + \rho s \end{aligned} \quad (5.13)$$

其中 ρ 是单位基本三次根,表达式为 $\rho = -\frac{1}{2} + i\frac{\sqrt{3}}{2}$ (这里要注意 $\rho^2 = -\frac{1}{2} - i\frac{\sqrt{3}}{2}$)。

做如下的替换,可以明显地简化计算。

$$\begin{aligned} p' &= \frac{p}{3} = -\frac{1}{9}a^2 + \frac{1}{3}b \\ q' &= \frac{q}{2} = \frac{1}{27}a^3 - \frac{1}{6}ab + \frac{1}{2}c \end{aligned} \quad (5.14)$$

这时判别式就变为

$$D = -108(p'^3 + q'^2) \quad (5.15)$$

令

$$D' = -\frac{D}{108} = p'^3 + q'^2 \quad (5.16)$$

则 r 和 s 表达式变成为

$$\begin{aligned} r &= \sqrt[3]{-q' + \sqrt{D'}} \\ s &= \sqrt[3]{-q' - \sqrt{D'}} \end{aligned} \quad (5.17)$$

和二次方程一样,根据判别式就可以判定三次方程的实根个数。如果 $D < 0$,也就是 D'

>0 , 公式(5.13)所给出的 x_1 的值就是方程(5.9)的惟一实根。

如果 $D = D' = 0$, 可以得到 $r = s$, 因此存在两个实数解, 其中一个为双根:

$$\begin{aligned} x_1 &= 2r \\ x_2, x_3 &= (\rho + \rho^2)r = -r \end{aligned} \quad (5.18)$$

对于其他情况(也就是 $D > 0$ 或 $D' < 0$ 的情况), 根据公式(5.13)可以得到三个不同的实根, 比较麻烦的是, 这时还需要使用复数来求解这些根。在这种情况下, 可以使用另外一种方法, 它不需要复数运算。在这种方法中用到了三角恒等式

$$4\cos^3\theta - 3\cos\theta = \cos 3\theta \quad (5.19)$$

该恒等式可以用 DeMoivre 定理证明(见本章的习题 1)。将 $x = 2m\cos\theta$ 代入方程(5.9), 并令 $m = \sqrt{-p/3}$, 可以得到

$$8m^3\cos^3\theta + 2pm\cos\theta + q = 0 \quad (5.20)$$

这里要注意, 为了使 D 为正, p 必须为负。用 $-3m^2$ 来替代 p , 并提取两项的系数 $2m^3$, 可以得到

$$2m^3(4\cos^3\theta - 3\cos\theta) + q = 0 \quad (5.21)$$

使用公式(5.19), 并求解 $\cos 3\theta$, 可以得到

$$\cos 3\theta = \frac{-q}{2m^3} = \frac{-q/2}{\sqrt{-p^3/27}} = \frac{-q'}{\sqrt{-p'^3}} \quad (5.22)$$

因为 $D' < 0$, 公式(5.16)表明 $q'^2 < -p'^3$, 这也保证了公式(5.22)右边的绝对值恒小于 1。根据反余弦的定义, 可以解得 θ 为

$$\theta = \frac{1}{3} \cos^{-1} \left(\frac{-q'}{\sqrt{-p'^3}} \right) \quad (5.23)$$

因此, 方程(5.9)的一个解为

$$x_1 = 2m\cos\theta = 2\sqrt{-p'}\cos\theta \quad (5.24)$$

因为对于任意整数 k 都有 $\cos(3\theta + 2\pi k) = \cos(3\theta)$, 因此有

$$\theta_k = \frac{1}{3} \cos^{-1} \left(\frac{-q'}{\sqrt{-p'^3}} \right) - \frac{2\pi}{3}k \quad (5.25)$$

k 取三个不同的值, 是以 3 为模的余数, 分别为 1, 2, 3。选择不同的 k 值, 就可以得到不同的 $\cos\theta_k$ 值。设 $k = \pm 1$, 就可以得到方程(5.9)其余的两个解:

$$\begin{aligned} x_2 &= 2\sqrt{-p'}\cos\left(\theta + \frac{2\pi}{3}\right) \\ x_3 &= 2\sqrt{-p'}\cos\left(\theta - \frac{2\pi}{3}\right) \end{aligned} \quad (5.26)$$

5.1.3 四次多项式

四次多项式的一般形式为

$$t^4 + at^3 + bt^2 + ct + d = 0 \quad (5.27)$$

其中已经做了必要的除法, 使得次数最高项的系数为 1。可以做如下的替换, 以去除 3 次项。令

$$t = x - \frac{a}{4} \quad (5.28)$$

则可以得到方程

$$x^4 + px^2 + qx + r = 0 \quad (5.29)$$

其中

$$\begin{aligned} p &= -\frac{3}{8}a^2 + b \\ q &= \frac{1}{8}a^3 - \frac{1}{2}ab + c \\ r &= -\frac{3}{256}a^4 + \frac{1}{16}a^2b - \frac{1}{4}ac + d \end{aligned} \quad (5.30)$$

一旦求得方程(5.29)的解 x , 减去 $a/4$, 就可以得到方程(5.27)的解 t 。
通过先求得三次方程

$$y^3 - \frac{p}{2}y^2 - ry + \frac{4rp - q^2}{8} = 0 \quad (5.31)$$

的解, 就可以得到四次方程的根。设 y 是这个方程的任意实数解, 如果 $q \geq 0$, 那么解这个四次方程就相当于解下面两个二次方程:

$$\begin{aligned} x^2 + x\sqrt{2y-p} + y - \sqrt{y^2-r} &= 0 \\ x^2 - x\sqrt{2y-p} + y + \sqrt{y^2-r} &= 0 \end{aligned} \quad (5.32)$$

如果 $q < 0$, 那么解四次方程相当于解下面的两个二次方程:

$$\begin{aligned} x^2 + x\sqrt{2y-p} + y + \sqrt{y^2-r} &= 0 \\ x^2 - x\sqrt{2y-p} + y - \sqrt{y^2-r} &= 0 \end{aligned} \quad (5.33)$$

5.1.4 牛顿-拉弗森迭代法

牛顿-拉弗森(Newton-Raphson)迭代法, 有时又称为牛顿法, 是求根的一种数值方法。使用牛顿法可以求得任意连续函数的根, 求解时要对由该连续函数及其导数所确定的公式进行迭代。

假设我们要求解图 5.1 中所示的函数的根。首先, 假定一个初始点 x_0 (后面会介绍怎样选这个初始值) 作为函数的猜想根, 曲线在点 $(x, f(x))$ 的切线斜率可以由 f 的导数给出, 这样可以给出该切线的方程为

$$y - f(x) = f'(x)(x - x_0) \quad (5.34)$$

观察图 5.1 可知, 切线与 x 轴相交于一点, 该交点比初始猜想根 x_0 更接近于 f 的实际根。在公式(5.34)中, 令 $y=0$ 并解方程, 可以得到根的求解迭代公式:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad (5.35)$$

其中, 用 x_{i+1} 重新标识 x , 用 x_i 重新标识 x_0 。反复运用此公式, 可以得到一个数列: $x_0, x_1,$

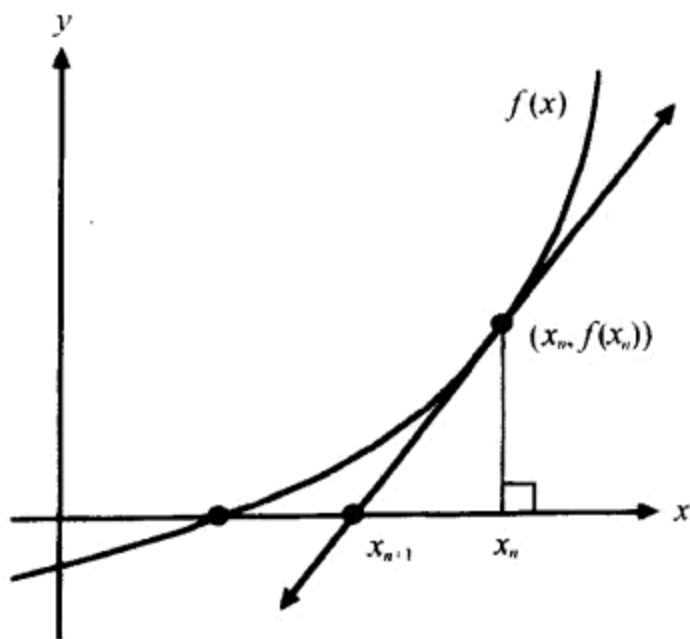


图 5.1 函数的切线与 x 轴的交点更趋近于函数的根

x_2, \dots 在正常情况下, 这些值趋近于 f 的根。

牛顿法收敛得非常快, 而且只需要很少的迭代次数就可以达到要求的精度。事实上, 可以证明牛顿法是二次收敛的, 也就是说每次迭代之后, 近似根有效数字的位数大约增加一倍。先作如下的设置, 可以证明此命题。假设

$$g(x) = \frac{f(x)}{f'(x)} \quad (5.36)$$

设 r 是收敛函数 f 的实际根, ϵ_i 是第 i 次近似计算所得值 x_i 与实际根 r 的差, 即

$$\epsilon_i = x_i - r \quad (5.37)$$

根据公式(5.35), 可写得

$$\epsilon_{i+1} = \epsilon_i - g(x_i) \quad (5.38)$$

用 $g(x_i)$ 的泰勒级数(参见附录 D)的前三项近似代替该函数, 可以得到

$$g(x_i) = g(r + \epsilon_i) \approx g(r) + \epsilon_i g'(r) + \frac{\epsilon_i^2}{2} g''(r) \quad (5.39)$$

$g(x)$ 的一阶和二阶导数分别为

$$g'(x) = 1 - \frac{f(x)f''(x)}{[f'(x)]^2}$$

$$g''(x) = \frac{2f(x)f'(x)[f''(x)]^2 - [f'(x)]^2[f(x)f'''(x) + f'(x)f''(x)]}{[f'(x)]^4} \quad (5.40)$$

因为 $f(r) = 0$, 所以在估算 r 值的时候, 上面的表达式会变得非常简单。当取值为 r 时, 函数 g 和它的两个导数分别为

$$g(r) = 0$$

$$g'(r) = 1$$

$$g''(r) = -\frac{f''(r)}{f'(r)} \quad (5.41)$$

将上式代入公式(5.39)可以得到

$$g(x_i) \approx \epsilon_i - \frac{\epsilon_i^2}{2} \frac{f''(r)}{f'(r)} \quad (5.42)$$

最后将上式再代入公式(5.38)得

$$\epsilon_{i+1} \approx \frac{\epsilon_i^2}{2} \frac{f''(r)}{f'(r)} \quad (5.43)$$

牛顿法并不保证一定收敛并得到解。在特殊的情况下, 如果初始猜想根选择在函数的导数为零的地方, 这时的切线为水平方向, 不会与 x 轴相交, 因此使用牛顿法就无法计算下去。初始猜想根必须选择在实际根附近, 这样才能保证收敛。当寻求一条光线与一个复杂物体的交点时, 通常可以先求这条光线与物体的包容边界面的交点(物体的包容边界面相对于物体本身要简单得多), 然后将该交点作为求解真正交点的初始猜想根。例如, 光线由 $P(t) = Q + tV$ 确定, 求光线与环形圆纹曲面的交点时, 可以先求得光线与环形圆纹曲面的外接长方体(包围盒)相交的 t 值, 然后将该 t 值作为求解光线和环形圆纹曲面交点的初始猜想根。

5.1.5 倒数与平方根的精化

当前大多数的图形硬件都能近似地计算一个数的倒数, 以及一个数的平方根倒数, 并且

至少保证数位的精度。比如,在 Nvidia 的 GeForce3 以及更先进的图形处理芯片上,对 OpenGL 功能进行扩展的 GL-vertex-program-NV 提供了对顶点的编程指令 RCP 和 RSQ,用这些指令可以近似地计算倒数和平方根倒数,并且利用牛顿法还可得到更高的精度。

计算下面函数的根可以求出 r 的倒数

$$f(x) = x^{-1} - r \quad (5.44)$$

因为 $f(1/r) = 0$,将该函数代入式(5.35),就可以得到

$$x_{n+1} = x_n - \frac{x_n^{-1} - r}{-x_n^{-2}} = x_n(2 - rx_n) \quad (5.45)$$

反复迭代此公式,可以求出 r 的高精度倒数,但要保证每次迭代 $x_i > 0$,因为函数 $f(x)$ 在 $x = 0$ 处存在着奇点。在第一次精化 x_1 的时候施加该条件,就可以求出初始近似值 x_0 存在的区间,因为 x_1 必须大于 0,所以必须满足

$$x_0(2 - rx_0) > 0 \quad (5.46)$$

由此可以推出的取值范围为

$$0 < x_0 < \frac{2}{r} \quad (5.47)$$

也就是说初始近似值肯定要小于 r 的倒数的 2 倍。

计算下面函数的正根可求得数值 r 的平方根的倒数:

$$f(x) = x^{-2} - r \quad (5.48)$$

将函数代到公式(5.35)中,可以得到

$$x_{n+1} = x_n - \frac{(x_n^{-2} - r)}{-2x_n^{-3}} = \frac{1}{2}x_n(3 - rx_n^2) \quad (5.49)$$

只要每次迭代都满足 $x_i > 0$,该数列就会收敛,因此初始近似值 x_0 必须满足

$$0 < x_0 < \sqrt{\frac{3}{r}} \quad (5.50)$$

因为 $\sqrt{r} = r(1/\sqrt{r})$,所以一旦在可接受的精度范围内计算出平方根倒数,就可以用简单的乘法算出 r 的平方根。

5.2 曲面交点

光线跟踪的核心问题是计算光线与曲面的交点。这一节讨论由

$$P(t) = Q + tV \quad (5.51)$$

确定的光线与常见对象(其他的对象留作练习)的交点的求法。除了三角形,交点的计算是在对象空间中进行的,在该空间中,对象的自然中心与原点、自然轴与坐标轴保持一致。在求与任意指向物体的交点时,首先要将光线变换到对象空间中,一旦求交完毕,将诸如交点、交点处的法向量等相关信息再变换回世界坐标空间中去。

5.2.1 光线与三角形相交

一个三角形是由它在空间中的三个顶点 P_0, P_1 和 P_2 的位置来描述的。通过先计算法向量 N ,可以确定三角形所在的平面,计算公式如下:

$$\mathbf{N} = (\mathbf{P}_1 - \mathbf{P}_0) \times (\mathbf{P}_2 - \mathbf{P}_0) \quad (5.52)$$

三角形到原点的带符号距离 D 通过计算 \mathbf{N} 与该平面上任一点的负点积得到, 这样可以选用顶点 \mathbf{P}_0 来构造 4D 平面向量 $\mathbf{L} = \langle \mathbf{N}, -\mathbf{N} \cdot \mathbf{P}_0 \rangle$ 。正如在 4.2.1 节所讨论的, 公式 (5.51) 所给出的光线与平面 \mathbf{L} 的交点处的 t 值为

$$t = -\frac{\mathbf{L} \cdot \mathbf{Q}}{\mathbf{L} \cdot \mathbf{V}} \quad (5.53)$$

当 $\mathbf{L} \cdot \mathbf{V} = 0$ 时, 线面不存在交点; 当 $\mathbf{L} \cdot \mathbf{V} \neq 0$ 时, 将 t 的值代入公式 (5.51), 就可以得到光线与三角形所在平面的交点 \mathbf{P} 。

现在要考虑的问题是如何判断点 \mathbf{P} 是否在三角形边的内侧。如果去掉 \mathbf{N} 的绝对值最大的分量所对应的坐标, 就可以将问题降至二维空间来考虑。如图 5.2 所示, 降维过程相当于将三角形投影到 xy, xz 或 yz 平面上。在这里假设被去掉的是 z 坐标, 在其他两种情况下被去掉的将分别是 x 和 y 坐标。

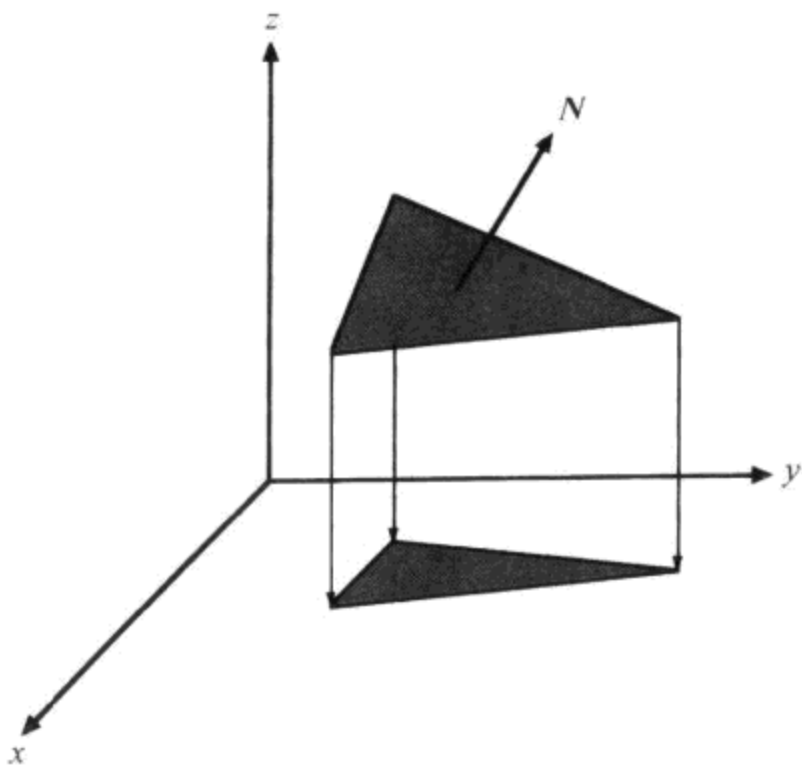


图 5.2 投影三角形到 xy, xz 或 yz 平面上

对三角形的每条边 i ($0 \leq i \leq 2$) 进行下面的二维差运算:

$$\begin{aligned} \mathbf{E} &= \mathbf{P}_{(i+1) \bmod 3} - \mathbf{P}_i \\ \mathbf{F} &= \mathbf{P}_{(i+2) \bmod 3} - \mathbf{P}_i \\ \mathbf{G} &= \mathbf{P} - \mathbf{P}_i \end{aligned} \quad (5.54)$$

这样做的本质意义是移动坐标系, 使得顶点 i 位于原点位置。如图 5.3 所示, 向量 \mathbf{E} 所表示的边有内侧和外侧之分, 将边向量旋转 90° (沿哪个方向并不重要), 可以构造出该边的法向量, 这里设 $\mathbf{N}_E = \langle -E_y, E_x \rangle$ 为得到的法向量。因为点 \mathbf{F} 一定位于边的内侧, 所以点积 $\mathbf{N}_E \cdot \mathbf{F}$ 的符号也就是任一内侧点与 \mathbf{N}_E 的点积的符号。因此, 如果 \mathbf{P} 点位于该边的内侧, 那么必须满足

$$(\mathbf{N}_E \cdot \mathbf{F})(\mathbf{N}_E \cdot \mathbf{G}) \geq 0 \quad (5.55)$$

对于任何点, 如果它都位于三角形的三条边的内侧, 那么它就在三角形之内。

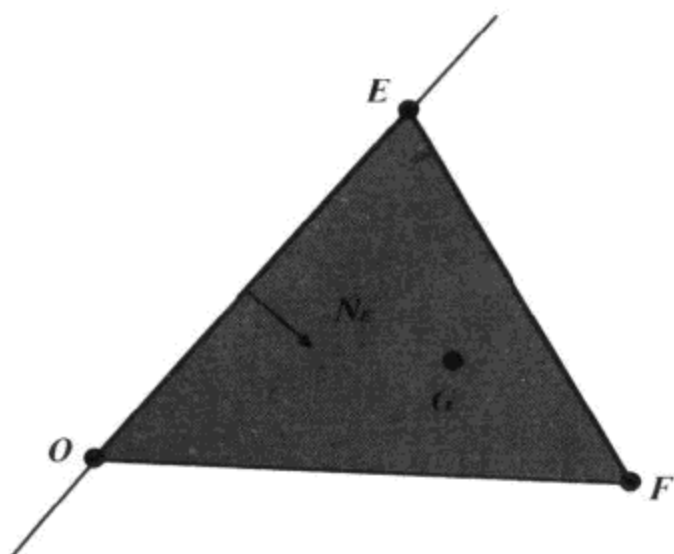


图 5.3 一点与边的法线的点积

5.2.2 光线与长方体相交

一个长方体可以用下面 6 个平面方程来表示：

$$\begin{aligned} x = 0 & \quad x = r_x \\ y = 0 & \quad x = r_y \\ z = 0 & \quad x = r_z \end{aligned} \quad (5.56)$$

其中 r_x, r_y 和 r_z 表示长方体的维。因为至少有三个平面与光线 \mathbf{V} 的方向相背, 所以需要考虑与光线的相交问题的平面最多有三个。通过检测光线 \mathbf{V} 的分量, 可以一次性地确定这些平面。例如, 如果 $V_x = 0$, 那么光线不可能与平面 $x = 0$ 或 $x = r_x$ 的任何一个相交, 因为 \mathbf{V} 与它们平行; 如果 $V_x > 0$, 那么不需考虑平面 $x = r_x$ 的相交情况, 因为它对于光线来说是长方体的背面; 同样, 如果 $V_x < 0$, 那么不需考虑平面 $x = 0$ 的相交情况。同样的原理适用于 \mathbf{V} 的 y 分量和 z 分量。

一旦找到了光线与平面的交点, 则必须确定该点是否位于长方体表面上, 这是通过检测交点平行于该平面的两个坐标分量来完成的。例如, t 对应的是公式(5.51)所确定的光线与平面 $x = r_x$ 的交点, 其值如下:

$$t = \frac{r_x - Q_x}{V_x} \quad (5.57)$$

为了保证位于长方体对应的表面上, 点 $\mathbf{P}(t)$ 的 t 和 z 坐标必须满足

$$\begin{aligned} 0 &\leq [\mathbf{P}(t)]_y \leq r_y \\ 0 &\leq [\mathbf{P}(t)]_z \leq r_z \end{aligned} \quad (5.58)$$

如果任一条件不满足, 那么在表面上不存在任何交点; 如果两个条件都满足, 则可以得到一个交点, 在这种情况下, 因为不会有其他更近的交点产生, 所以不需要对其他平面进行检测。

5.2.3 光线与球体相交

一个以原点为球心, r 为半径的球面可以用下面的方程来表示:

$$x^2 + y^2 + z^2 = r^2 \quad (5.59)$$

用公式(5.51)给出的光线的分量代替 x, y 和 z 可以得到

$$(Q_x + tV_x)^2 + (Q_y + tV_y)^2 + (Q_z + tV_z)^2 = r^2 \quad (5.60)$$

将平方展开且合并 t 的同类项,可以得到下面的二次方程:

$$(V_x^2 + V_y^2 + V_z^2)t^2 + 2(Q_xV_x + Q_yV_y + Q_zV_z)t + Q_x^2 + Q_y^2 + Q_z^2 - r^2 = 0 \quad (5.61)$$

公式(5.2)中的系数 a, b 和 c 可以用向量 \mathbf{Q} 和 \mathbf{V} 表示如下:

$$\begin{aligned} a &= V^2 \\ b &= 2(\mathbf{Q} \cdot \mathbf{V}) \\ c &= Q^2 - r^2 \end{aligned} \quad (5.62)$$

通过计算判别式 $D = b^2 - 4ac$, 可以判断光线是否和球面相交。如图 5.4 所示, 如果 $D < 0$, 则不相交; 如果 $D = 0$, 则光线与球面相切; 如果 $D > 0$, 则有两个不同的交点。如果光线与球面有两个不同的交点, 光源点为 \mathbf{Q} , 那么越接近 \mathbf{Q} 的交点所对应的 t 值越小, t 值计算公式为

$$t = \frac{-b - \sqrt{D}}{2a} \quad (5.63)$$

之所以选择该式来计算 t 值, 是因为事先保证 a 肯定为正。

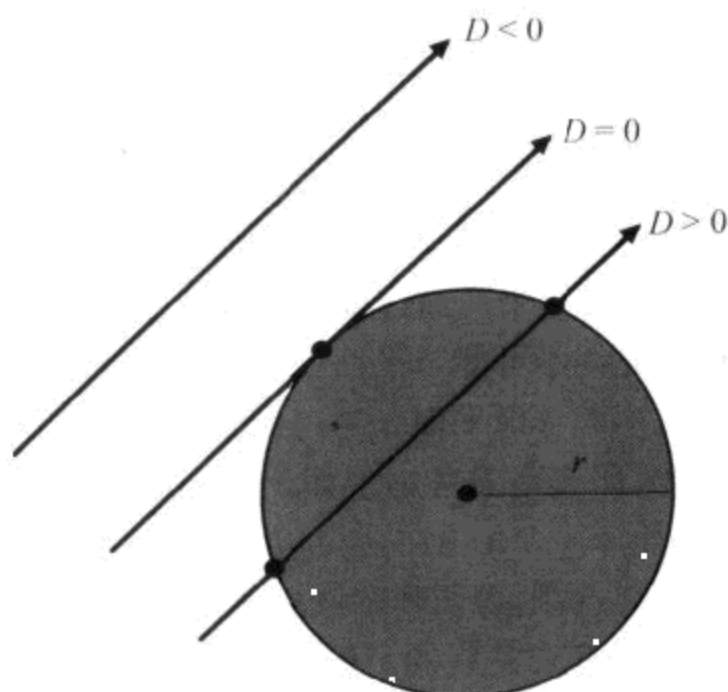


图 5.4 判别式指出了光线与球体是否相交

用下面的公式代替等式(5.59), 可以求解光线和椭球面的交点:

$$x^2 + m^2y^2 + n^2z^2 = r^2 \quad (5.64)$$

其中 m 是 y 半轴长度与 x 半轴长度的比值, n 是 z 半轴长度与 x 半轴长度的比值。将光线的分量代入到公式中可以得到另外一个二次多项式, 其中系数为

$$\begin{aligned} a &= V_x^2 + m^2V_y^2 + n^2V_z^2 \\ b &= 2(Q_xV_x + m^2Q_yV_y + n^2Q_zV_z) \\ c &= Q_x^2 + m^2Q_y^2 + n^2Q_z^2 - r^2 \end{aligned} \quad (5.65)$$

同样, 用判别式可以判断出是否存在交点, 如果存在, 交点参数 t 由公式(5.63)给出。

5.2.4 光线与圆柱面相交

如图 5.5 所示,沿 x 轴半径为 r ,沿 y 轴半径为 s ,高度为 h ,底部在 xy 平面上,并且圆心在原点的椭圆柱面的侧表面可以表示为

$$x^2 + m^2 y^2 = r^2 \quad 0 \leq z \leq h \quad (5.66)$$

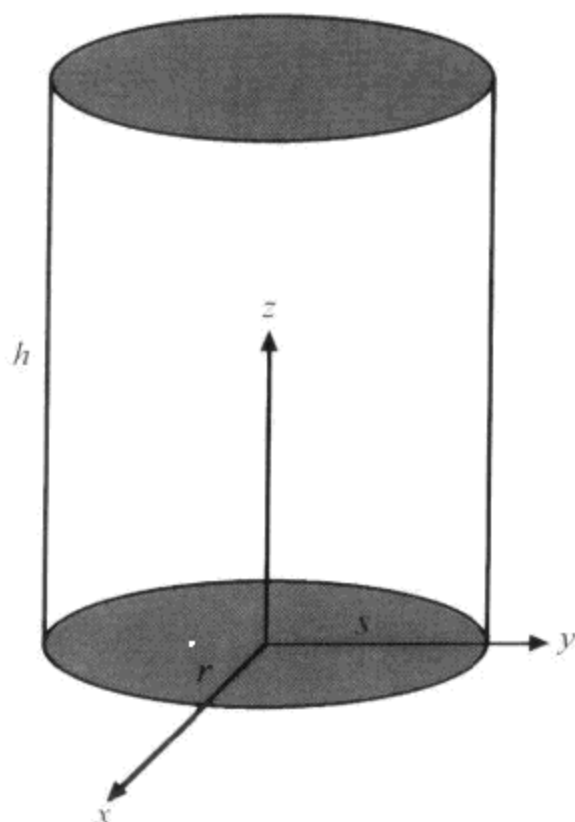


图 5.5 椭圆柱的物体空间

其中 $m = r/s$ 。如果 $r = s$,那么椭圆柱面就是圆柱面,并且 $m = 1$ 。用公式(5.51)给出的光线的分量分别代替 x 和 y ,可以得到

$$(Q_x + tV_x)^2 + m^2(Q_y + tV_y)^2 = r^2 \quad (5.67)$$

展开平方,并合并 t 的同类项,可以得到下面的二次方程:

$$(V_x^2 + m^2 V_y^2)t^2 + 2(Q_x V_x + m^2 Q_y V_y)t + Q_x^2 + m^2 Q_y^2 - r^2 = 0 \quad (5.68)$$

和球面一样,根据判别式可以判断是否存在交点。这个方程的解给出了光线与以 z 轴为轴心的无限圆柱面相交时的 t 值,所以必须检验交点的 z 坐标是否满足 $0 \leq z \leq h$ 。

5.2.5 光线与环形圆纹曲面相交

如图 5.6 所示,环形圆纹曲面的横截面有两个半径,分别为主半径 r_1 和辅半径 r_2 。

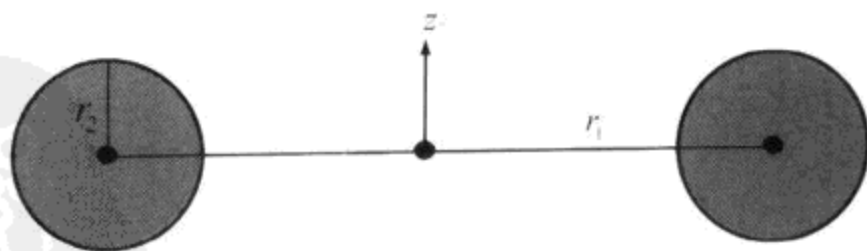


图 5.6 环形圆纹曲面的横截面

半径为 r_1 的圆位于 xy 平面,它表示半径为 r_2 并且垂直于 r_1 的另外一个圆的中心,后者绕 z 轴进行旋转。描述该旋转圆的方程为

$$s^2 + z^2 = r_2^2 \quad (5.69)$$

其中 s 值是在 xy 平面上点到主圆的距离,即

$$s = \sqrt{x^2 + y^2} - r_1 \quad (5.70)$$

将该式代入公式(5.69)并展开平方可得

$$x^2 + y^2 + z^2 + r_1^2 - r_2^2 - 2r_1\sqrt{x^2 + y^2} = 0 \quad (5.71)$$

将根号项单独移动到公式的右边,两边同时平方就可得到下面的环形圆纹曲面方程:

$$(x^2 + y^2 + z^2 + r_1^2 - r_2^2)^2 = 4r_1^2(x^2 + y^2) \quad (5.72)$$

用光线的三个分量代替公式(5.51)中的 x 、 y 和 z ,可以得到

$$\begin{aligned} & [(Q_x + tV_x)^2 + (Q_y + tV_y)^2 + (Q_z + tV_z)^2 + r_1^2 - r_2^2]^2 \\ & = 4r_1^2[(Q_x + tV_x)^2 + (Q_y + tV_y)^2] \end{aligned} \quad (5.73)$$

经过代数简化,这个等式可以表示为四次方程

$$at^4 + bt^3 + ct^2 + dt + e = 0 \quad (5.74)$$

其中

$$\begin{aligned} a &= V^4 \\ b &= 4V^2(\mathbf{Q} \cdot \mathbf{V}) \\ c &= 2V^2(Q^2 + r_1^2 - r_2^2) - 4r_1^2(V_x^2 + V_y^2) + 4(\mathbf{Q} \cdot \mathbf{V})^2 \\ d &= 8r_1^2Q_zV_z + 4(\mathbf{Q} \cdot \mathbf{V})(Q^2 - r_1^2 - r_2^2) \\ e &= Q_x^4 + Q_y^4 + Q_z^4 + (r_1^2 - r_2^2)^2 + \\ & \quad 2[Q_x^2Q_y^2 + Q_z^2(r_1^2 - r_2^2) + (Q_x^2 + Q_y^2)(Q_z^2 - r_1^2 - r_2^2)] \end{aligned} \quad (5.75)$$

方程两边除以 a 后,左边第一项的系数为 1,这样该方程可以使用 5.1.3 节给出的方法进行求解。

5.3 法向量的计算

有时用隐函数 $f(x, y, z)$ 能够很方便地表示一个平面,在平面上任一点 (x, y, z) 该函数数值为零,而在平面之外函数值不为零。比如,一个表示椭圆面的函数为

$$f(x, y, z) = \frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} - 1 \quad (5.76)$$

这里使用的就是隐函数表示法,从中可以导出曲面上任一点处法向量的公式。

假设 $f(x, y, z)$ 表示一个曲面 S ,则 $f(x, y, z) = 0$ 对曲面 S 上的所有点都成立。设 C 是位于曲面 S 上并且由可微函数 $x(t)$ 、 $y(t)$ 和 $z(t)$ 定义的一条曲线。曲线 C 在点 $(x(t), y(t), z(t))$ 的切向量 \mathbf{T} 表示为

$$\mathbf{T} = \left\langle \frac{d}{dt}x(t), \frac{d}{dt}y(t), \frac{d}{dt}z(t) \right\rangle \quad (5.77)$$

由于曲线 C 在曲面 S 上,所以 \mathbf{T} 也与曲面 S 相切。同样,由于对任意的 t 值都有 $f(x(t), y(t), z(t)) = 0$,所以在曲线 C 上任一点处 $df/dt = 0$ 都成立。根据链式法则,可

以写成

$$0 = \frac{df}{dt} = \frac{\partial f}{\partial x} \frac{dx}{dt} + \frac{\partial f}{\partial y} \frac{dy}{dt} + \frac{\partial f}{\partial z} \frac{dz}{dt} = \left\langle \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right\rangle \cdot \mathbf{T} \quad (5.78)$$

因为与 \mathbf{T} 的点积总为零,所以向量 $\langle \partial f/\partial x, \partial f/\partial y, \partial f/\partial z \rangle$ 肯定是平面 S 的法向量。这个向量称为 f 在点 $\langle x, y, z \rangle$ 处的梯度,通常记为 $\nabla f(x, y, z)$,其中符号 ∇ 是 del 操作符,定义为

$$\nabla = \mathbf{i} \frac{\partial}{\partial x} + \mathbf{j} \frac{\partial}{\partial y} + \mathbf{k} \frac{\partial}{\partial z} \quad (5.79)$$

现在可以用一个公式来表示由 $f(x, y, z) = 0$ 确定的曲面的法向量 \mathbf{N} 为

$$\mathbf{N} = \nabla f(x, y, z) \quad (5.80)$$

继续(5.76)式给出的例子,可以用下面的公式给出椭圆面的法向量:

$$\mathbf{N} = \left\langle \frac{2x}{a^2}, \frac{2y}{b^2}, \frac{2z}{c^2} \right\rangle \quad (5.81)$$

5.4 反射向量和折射向量

当一束光线射到物体的表面时,它的一部分能量被物体表面吸收,一部分能量被表面反射,其余部分的能量则穿过物体本身,将在第6章中详细讨论这种交互过程。这一节介绍如何计算光线投射到光滑或透明曲面所产生的反射和折射。

5.4.1 反射向量的计算

光滑表面(比如镜面)对光线的反射方向遵循一个简单的原理,即入射角等于反射角。如图5.7所示,法向量 \mathbf{N} 与指向入射光的 \mathbf{L} 的夹角,等于法向量 \mathbf{N} 与反射光方向 \mathbf{R} 的夹角。

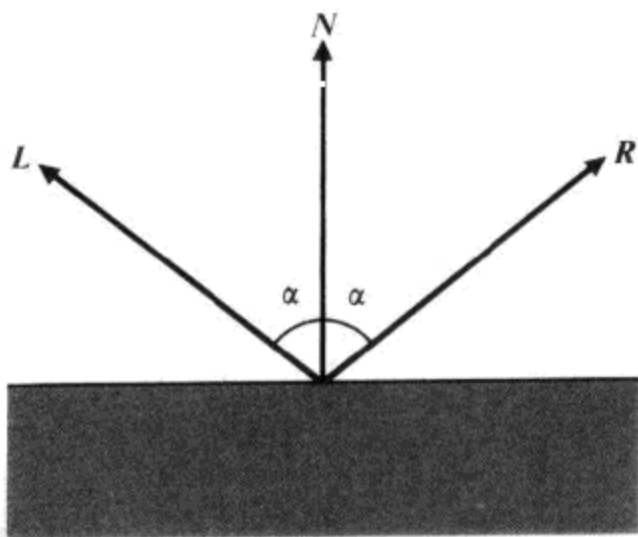


图 5.7 入射角等于反射角

这里假设向量 \mathbf{N} 和 \mathbf{L} 已经单位化。为了导出用指向入射光的 \mathbf{L} 和法向量 \mathbf{N} 来计算反射光方向 \mathbf{R} 的公式,首先要计算出 \mathbf{L} 的垂直于法向量的分量:

$$\text{perp}_{\mathbf{N}}\mathbf{L} = \mathbf{L} - (\mathbf{N} \cdot \mathbf{L})\mathbf{N} \quad (5.82)$$

如图 5.8 所示, 向量 R 到 L 的距离是它到法向量 N 的投影距离的两倍。这样可以给出 R 的计算公式为

$$\begin{aligned} R &= L - 2\text{perp}_N L \\ &= L - 2[L - (N \cdot L)N] \\ &= 2(N \cdot L)N - L \end{aligned} \quad (5.83)$$

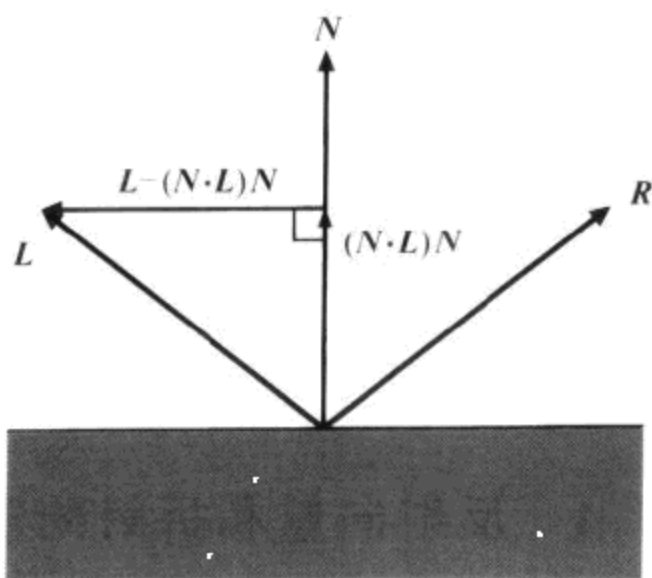


图 5.8 求反射向量 R

5.4.2 折射向量的计算

透明表面具有一个特殊的性质, 称为折射系数。根据 Snell 定律, 入射角 θ_L 和透射角 θ_T (如图 5.9 所示) 之间的关系可以表示为

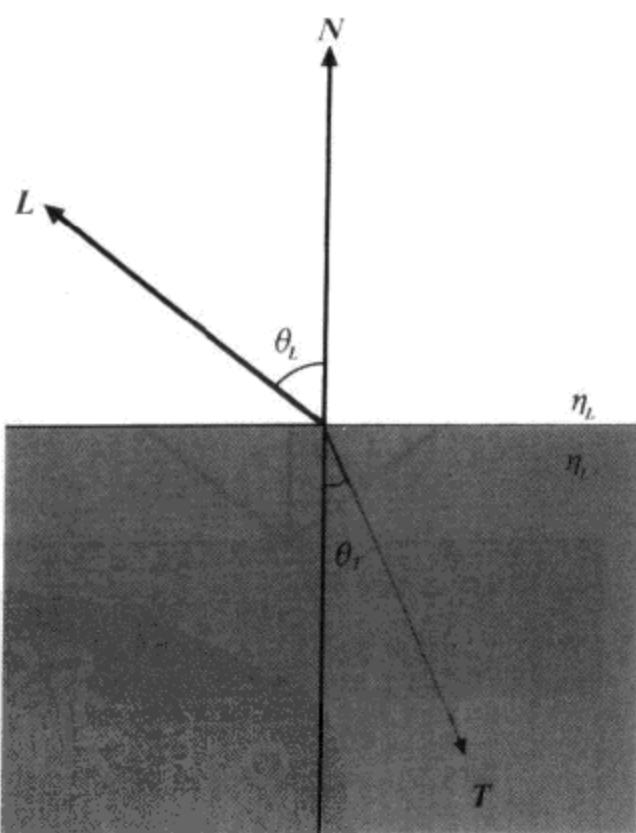


图 5.9 入射角 θ_L 与透射角 θ_T 之间的关系

$$\eta_L \sin \theta_L = \eta_T \sin \theta_T \quad (5.84)$$

其中 η_L 是光线离开媒质的折射系数, 而 η_T 是光线进入媒质的折射系数。空气的折射系数通常为 1。两种媒质的折射系数的差别越大, 在界面处所产生的弯曲效果就越明显。

这里假设法向量 \mathbf{N} 和指向入射光的 \mathbf{L} 已经单位化了。把透射光传输的方向 \mathbf{T} 表示为平行于法向量的分量和垂直于法向量的分量。如图 5.10 所示, \mathbf{T} 平行于法向量的分量可以简单地表示为 $-\mathbf{N} \cos \theta_T$, \mathbf{T} 垂直于法向量的分量可表示为 $-\mathbf{G} \sin \theta_T$, 其中向量 \mathbf{G} 是平行于 $\text{perp}_{\mathbf{N}} \mathbf{L}$ 的单位向量。由于 \mathbf{L} 是单位向量, $\|\text{perp}_{\mathbf{N}} \mathbf{L}\| = \sin \theta_L$, 所以有

$$\mathbf{G} = \frac{\text{perp}_{\mathbf{N}} \mathbf{L}}{\sin \theta_L} = \frac{\mathbf{L} - (\mathbf{N} \cdot \mathbf{L}) \mathbf{N}}{\sin \theta_L} \quad (5.85)$$

现在, 可以把折射向量 \mathbf{L} 表示为

$$\begin{aligned} \mathbf{T} &= -\mathbf{N} \cos \theta_T - \mathbf{G} \sin \theta_T \\ &= -\mathbf{N} \cos \theta_T - \frac{\sin \theta_T}{\sin \theta_L} [\mathbf{L} - (\mathbf{N} \cdot \mathbf{L}) \mathbf{N}] \end{aligned} \quad (5.86)$$

利用公式(5.84), 可以用 η_L / η_T 代替正弦的商, 即

$$\mathbf{T} = -\mathbf{N} \cos \theta_T - \frac{\eta_L}{\eta_T} [\mathbf{L} - (\mathbf{N} \cdot \mathbf{L}) \mathbf{N}] \quad (5.87)$$

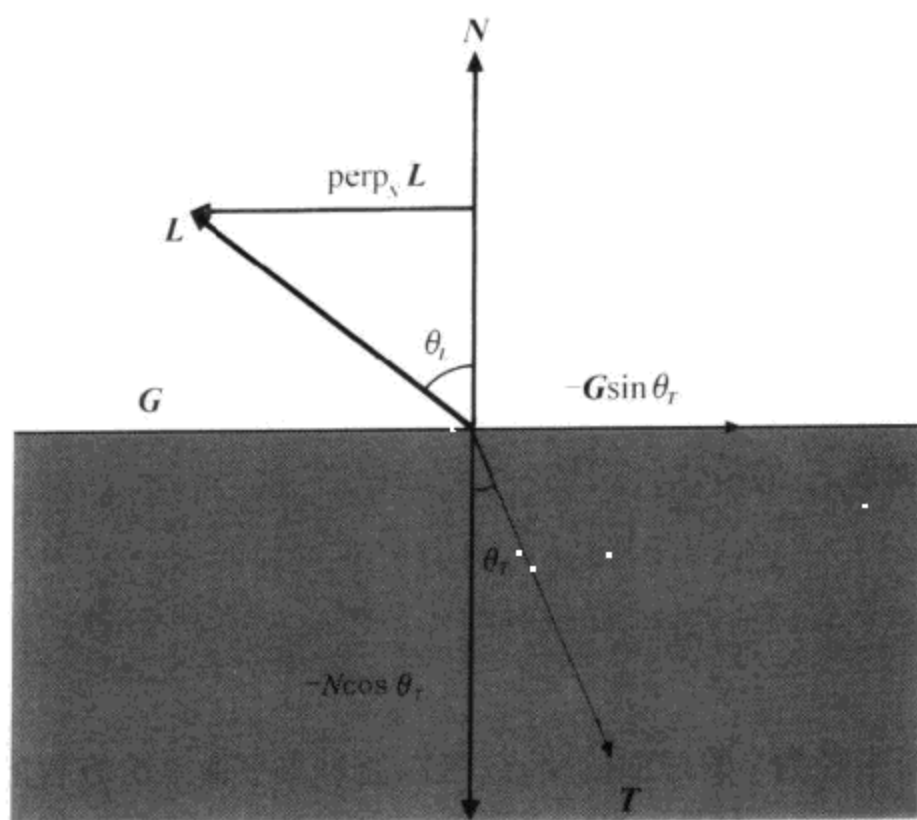


图 5.10 折射向量 \mathbf{T} 用平行于法向量 \mathbf{N} 的分量和垂直于 \mathbf{N} 的分量来表示

用 $\sqrt{1 - \sin^2 \theta_T}$ 代替 $\cos \theta_T$, 然后根据公式(5.84), 用 $(\eta_L / \eta_T) \sin \theta_T$ 代替 $\sin \theta_T$, 这样可以得到

$$\mathbf{T} = -\mathbf{N} \sqrt{1 - \frac{\eta_L^2}{\eta_T^2} \sin^2 \theta_L} - \frac{\eta_L}{\eta_T} [\mathbf{L} - (\mathbf{N} \cdot \mathbf{L}) \mathbf{N}] \quad (5.88)$$

用 $1 - \cos^2 \theta_L - (\mathbf{N} \cdot \mathbf{L})^2$ 代替 $\sin^2 \theta_L$, 最终可以得到

$$T = \left[\frac{\eta_L}{\eta_T} \mathbf{N} \cdot \mathbf{L} - \sqrt{1 - \frac{\eta_L^2}{\eta_T^2} [1 - (\mathbf{N} \cdot \mathbf{L})^2]} \right] \mathbf{N} - \frac{\eta_L}{\eta_T} \mathbf{L} \quad (5.89)$$

如果 $\eta_L > \eta_T$, 公式(5.89)中根号里的值可能为负, 当光线从有较高折射系数的媒质以较大的人射角进入较低折射系数的媒质时, 就会出现这种情形。需要注意的是, 公式(5.89)仅当 $\sin\theta_L \leq \eta_T/\eta_L$ 时才成立。如果根号里的值为负, 就会出现一种称为全反射的现象, 此时光线不发生折射, 只根据公式(5.83)在媒质内部发生反射。

5.5 本章小结

分析求根法

二次方程 $at^2 + bt + c = 0$ 的解由下面的二次等式表示:

$$t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

三次和四次方程, 同样也可以分析地求解。

数值求根法

使用牛顿法可对函数 $f(x)$ 进行数值法求解, 牛顿法使用迭代公式

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

来精化近似解 x_n 。

对于数字 r 的倒数 x_n 的精化公式是

$$x_{n+1} = x_n(2 - rx_n)$$

对于数字 r 的平方根倒数 x_n 的精化公式是

$$x_{n+1} = \frac{1}{2} x_n(3 - rx_n^2)$$

光线和球体相交

光线 $P(t) = Q + tV$ 与半径为 r 的球面的交点, 可以通过求下面二次方程的解得到。

$$V^2 t^2 + 2(Q \cdot V)t + Q^2 - r^2 = 0$$

法向量的计算

在函数 $f(x, y, z) = 0$ 所定义的曲面上的点 $\langle x, y, z \rangle$ 处的法向量定义为

$$\mathbf{N} = \nabla f(x, y, z)$$

反射向量的计算

向量 L 对应于法向量 N 的反射向量 R 为

$$\mathbf{R} = 2(\mathbf{N} \cdot \mathbf{L})\mathbf{N} - \mathbf{L}$$

透射向量的计算

当光线离开折射系数为 η_L 的媒质, 进入折射系数为 η_T 的媒质时, 透射方向 \mathbf{T} 可以表示为

$$\mathbf{T} = \left[\frac{\eta_L}{\eta_T} \mathbf{N} \cdot \mathbf{L} - \sqrt{1 - \frac{\eta_L^2}{\eta_T^2} [1 - (\mathbf{N} \cdot \mathbf{L})^2]} \right] \mathbf{N} - \frac{\eta_L}{\eta_T} \mathbf{L} \quad (5.89)$$

其中, \mathbf{L} 是指向入射光线的方向, \mathbf{N} 是表面法线。

习 题

1. 用 DeMoivre 定理(即 $e^{i\alpha} = \cos\alpha + i\sin\alpha$)证明下面三角恒等式:

$$4\cos^3\theta - 3\cos\theta = \cos(3\theta)$$

提示: 使等式 $(e^{i\theta})^3 = e^{(3\theta)i}$ 两边的实部相等。

2. 用牛顿-拉弗森迭代法, 求方程 $f(x) = \ln x + x - 7$ 的近似根。
 3. 用牛顿-拉弗森迭代法, 求出能精化数字 r 的 P 次根的近似值 x_n 的通用公式。
 4. 推导一个多项式, 它的根是直线 $\mathbf{P}(t) = \mathbf{Q} + t\mathbf{V}$ 与一个圆锥面的交点。如图 5.11 所示, 该圆锥面的底半径为 r , 高为 h , 底的中心位于 xy 平面的原点。

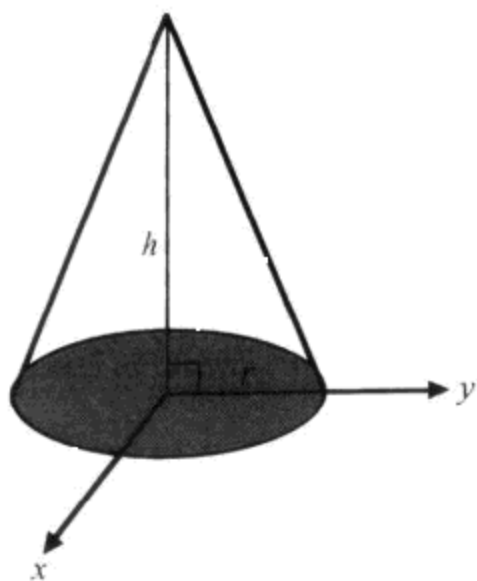


图 5.11 习题 4 用到的圆锥面

5. 计算抛物面 $f(x, y, z) = 2x^2 + 3y^2 - z = 0$ 在点 $(-1, 2, 14)$ 处的表面单位法向量。

6. 假定坐标变换定义为
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = M \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

其中 M 是一个可逆的 3×3 矩阵。带撇号的坐标系中 del 的运算符 ∇' 定义为

$$\nabla' = \left\langle \frac{\partial}{\partial x'}, \frac{\partial}{\partial y'}, \frac{\partial}{\partial z'} \right\rangle$$

证明: $\nabla' = (\mathbf{M}^{-1})^T \nabla$

[提示: 将不带撇号的坐标看成是三个带撇号坐标的函数, 如下表示:

$$x = x(x', y', z')$$

$$y = y(x', y', z')$$

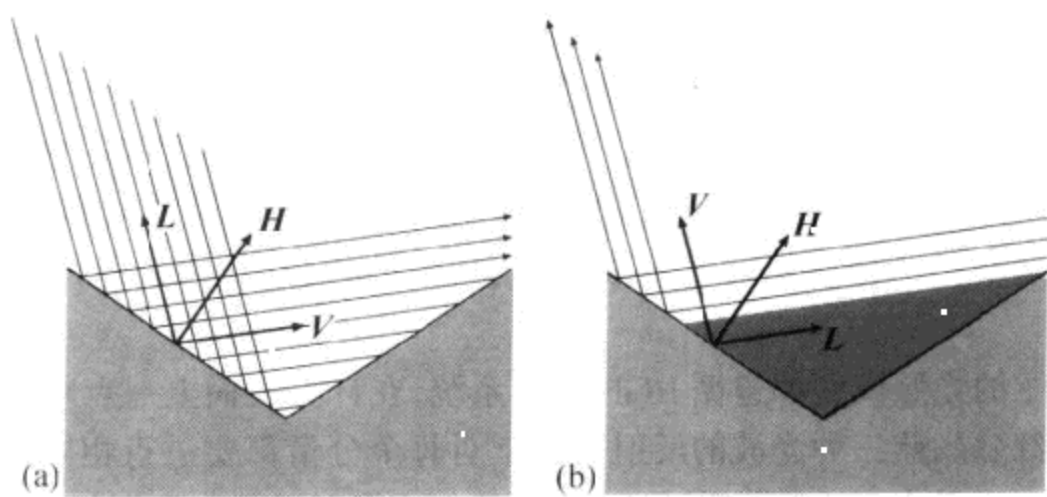
$$z = z(x', y', z')$$

并使用偏微分的链式法则,如对于 x 坐标有 $\frac{\partial}{\partial x'} = \frac{\partial x}{\partial x'} \frac{\partial}{\partial x} + \frac{\partial y}{\partial x'} \frac{\partial}{\partial y} + \frac{\partial z}{\partial x'} \frac{\partial}{\partial z}$

7. 两个媒质间的分界面上的临界角就是发生全反射的最小入射角。试求光线穿过水到达空气的临界角。水的折射系数是 1.33, 空气的折射系数是 1.00。



第 6 章 光 照



这一章讲述有关表面光照的数学问题。光照过程,通常称为明暗处理,要求确定构成表面的每个像素反射到观察者眼中的光的颜色。这些颜色一方面取决于照射到表面上的光源的特性,另一方面也依赖于表面本身的反射特性。

光和表面之间的相互作用是一个复杂的物理过程。当撞击到物质的表面时,光子可能被吸收、反射或投射。如果使用当今所知的所有物理知识对这个交互过程进行建模,将要花费难以承受的计算开支,因此,应该建立一种简化模型,尽可能地接近所期望的表面特性。本章首先介绍一些简单的模型,这些模型的计算效率很高,产生的结果能够为我们所接受,所以得到了广泛的应用,当然这些模型缺少物理上的精确性。然后,将讨论计算开支更大但更能准确地反映光和表面之间物理交互过程的建模技术。

6.1 RGB 颜色系统

一个描述表面反射光的精确模型,应该能够表示可见光谱中各种波长的光。然而,对于大多数计算机显示器来说,颜色信息的显示只是依靠三种光波的组合:红、绿、蓝。将这样的红、绿、蓝组合系统称为 RGB 颜色系统。对于波长介于三种颜色之间的光,可以通过将三种主颜色按一定的比例进行混合来模拟,比如,黄色可以通过混合相同比例的红色和绿色而得到。对于那些由不止一种波长的光组成的颜色,比如褐色,也可以使用 RGB 颜色系统来模拟。

本章所介绍的光照模型都遵循 RGB 颜色系统,在计算表面上一点对光的反射时,要同时计算该点对红、绿、蓝三种光波的反射。由于对每个分量都要进行相同的计算,所以可以使用一种颜色三元组来表达数学公式,这种颜色三元组可以简称为颜色。

颜色可以用红、绿、蓝三元组来表示,每个分量都在 0 到 1 的区间取值。这样,颜色既能够代表光的光谱成分(决定了眼睛所看到的颜色),同时也能够代表光的强度。这里使用下标 r, g, b 来表示颜色 C 的分量,所以一种颜色可以写作 $C = \langle C_r, C_g, C_b \rangle$ 。

颜色 C 可以通过乘以一个比例系数而变成一个新的颜色

$$sC = \langle sC_r, sC_g, sC_b \rangle \quad (6.1)$$

颜色的加法和乘法是通过其分量的加法和乘法实现的。也就是,对于颜色 C 和 D 有

$$\begin{aligned} C + D &= \langle C_r + D_r, C_g + D_g, C_b + D_b \rangle \\ CD &= \langle C_r D_r, C_g D_g, C_b D_b \rangle \end{aligned} \quad (6.2)$$

颜色的乘法,也称为调制,可以是一个颜色与另一个颜色相乘,也可以是颜色乘以一个数量系数。渲染三角形时,三角形上的每个像素的颜色通常由多个光源的颜色组合来决定。三角面上像素的颜色通常来自两个颜色的乘积,一个是在纹理映射中得到的颜色,另一个是对三角形顶点颜色的插值。这种情况下,称用顶点颜色对纹理颜色进行调制。

6.2 光源

所计算的表面上任何一点的颜色是照射到表面上的所有光源的总和。3D 图形系统支持的光源的可以归纳为四种标准类型:环境光、定向光、点光源和聚焦光。这一节介绍每一种类型的光源及它们是如何使空间中的点发出光亮的。

6.2.1 环境光

在某一区域出现的环境光是一种低强度的光,它是由光线经周围环境中所有的临近表面多次反射后形成的。利用环境光可以近似地描述区域的大概亮度,这样就避免了场景中所有物体间相互反射的复杂计算。

环境光可以认为来自四面八方,并且在各个方向具有相同的强度,能够均匀地照亮物体的每个部分。在一个场景中,环境光的颜色 A 通常是一个常量,当然有时也可以是一个关于空间位置的函数,比如,对于现实世界中单一的区域,可以对该区域用规则网格进行划分,然后进行环境光采样,最后将采样结果记录到一个纹理映射图中。

6.2.2 定向光源

定向光源是一个无穷光源,它在单一的方向从无限远处发射光线。定向光源可以用来对诸如太阳光的光源进行建模,可以认为这种光源发出的光是平行的。由于定向光源在空间中没有位置,具有无限的射程,并且发射的光强度不像点光源和聚焦光那样随着距离的增加而减弱。

6.2.3 点光源

点光源是一种在空间中的某个点上向各个方向等强度地发射光线的光源。根据平方反比律,光线的强度随距离的增加而自然减弱。在 OpenGL 和 Direct3D 中都对这种原理进行了实现,允许使用二次多项式的倒数来控制点光源所发出的光线的强度。

假设一个点光源位于 P 点,到达空间一点 Q 的光强 C 可以通过下面的公式计算:

$$C = \frac{1}{k_c + k_l d + k_q d^2} C_0 \quad (6.3)$$

其中, C_0 是光的颜色, d 是光源和 Q 之间的距离(即 $d = \|P - Q\|$), 常数 k_c, k_l 和 k_q 分别是衰减常量、线性衰减常量和二次衰减常量。

6.2.4 聚焦光源

聚焦光源和点光源类似,不同的是聚焦光源具有主辐射方向。聚焦光源和点光源一样,强度随着距离的增加而衰减,除此之外,它还受到另外一种衰减因素的影响,即聚光灯效应。

假设在点 P 有一聚焦光源,方向为 U , 到达空间一点 Q 的光强 C 可以用下面的公式计算:

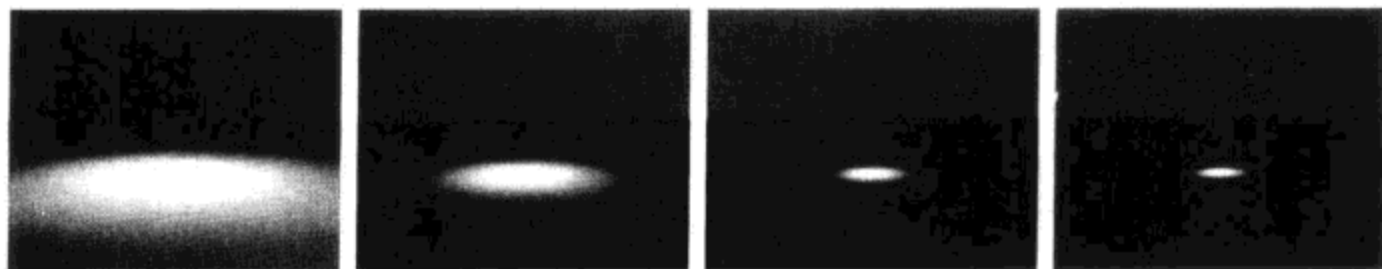
$$C = \frac{\max\{-U \cdot L, 0\}^p}{k_c + k_l d + k_q d^2} C_0 \quad (6.4)$$

其中 C_0 是光的颜色, d 是光源和 Q 的距离, k_c, k_l 和 k_q 是衰减常数, L 是从 Q 指向光源的单位方向向量

$$L = \frac{P - Q}{\|P - Q\|} \quad (6.5)$$

指数 p 控制聚焦光的聚焦方式。如图 6.1 所示,较大的指数 p 对应于高聚焦度的聚焦光,光强从聚焦中心向外出现急剧衰减;反之,较小的指数 p 则对应聚焦度不高的光柱。当

$U = -L$ 时, 聚焦光的光强最大, 当 U 和 $-L$ 之间的夹角增大时, 光强将逐渐衰减。当一个点的 U 和 $-L$ 之间的夹角大于 90° 时, 聚焦光源发出的光线不会到达该点。



注: 从左到右, 照射地面的聚焦光的聚焦指数分别是 2, 10, 50, 100。

图 6.1 聚焦光指数控制了聚焦光的聚焦程度

6.3 漫反射光

漫反射面是这样一种表面: 当光照射到它上面的一点时, 光将在任意方向上散射。一般的效果就是光的某种颜色 (就是表面漫反射的颜色) 向各个方向均匀地反射。这种反射称为 Lambertian 反射, 因为光在各个方向上的反射是相等的, 所以兰伯特反射的观察效果与观察者所处的位置无关。

如图 6.2 所示, 对于横截面积为 A 的光束, 只有当它的入射方向与表面垂直时, 照射面积才为 A , 随着法向量和光线方向之间夹角的增加, 被光束照射的面积也随之增加。如果法向量与光线方向的夹角为 θ , 则该光束照射的面积为 $A/\cos\theta$, 也说明单位面积上的光强是以 $\cos\theta$ 为系数进行衰减的。

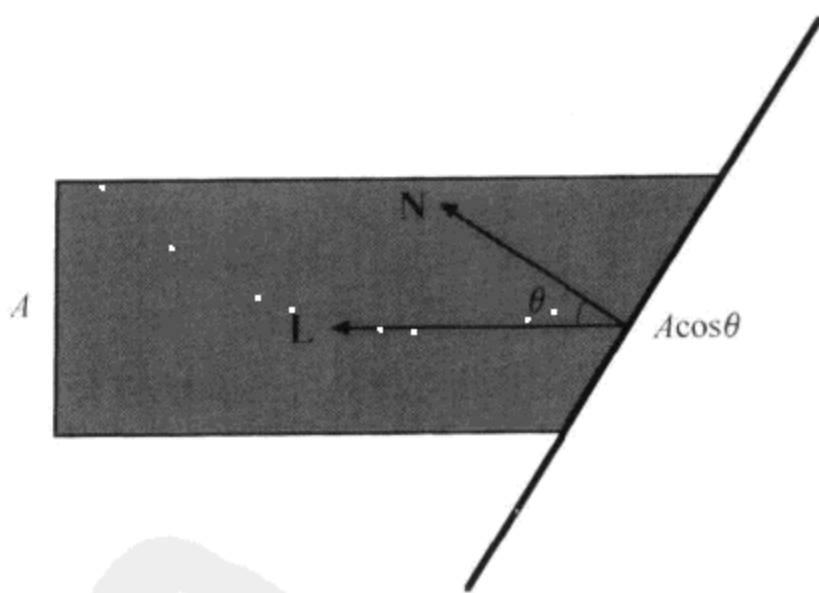


图 6.2 光束照射的面积随夹角 θ 的增加而增加

注意: 光束所照射的面积随着表面法线与光的入射方向的夹角的增加而增加, 而单位面积的入射光的光强减少。

$\cos\theta$ 的值可由法向量 N 与指向光源的单位方向向量 L 的点积得到, 如果点积为负说明该表面是背向光源的, 根本不会被光线照到, 这时可以在光照计算中用零来代替点积的结果。

现在可以建立一个公式,用它来计算表面上给定点 Q 反射到观察者眼中光的颜色 K 。在该公式中, C_i 是 n 种光源中每种光照射到点 Q 的光强,它对于定向光源来说是个常量,对于点光源和聚焦光源则由公式(6.3)和(6.4)给出。反射光由表面漫反射颜色 D 进行调制。把 n 个光源对该点的作用量相加,并考虑环境光强 A ,这样可以将光照公式的漫射分量表示为

$$K_{\text{diffuse}} = DTA + DT \sum_{i=1}^n C_i \max\{N \cdot L_i, 0\} \quad (6.6)$$

其中单位向量 L_i 从点 Q 指向第 i 个光源。

6.4 纹理映射

如图 6.3 所示,在表面上使用一个或多个纹理图可以更详细地表现表面的细节。对于表面上的每个点,可以在每个纹理图中查出其对应的纹理像素,并且可以将纹理像素与光照公式以某种方式结合起来。在最简单的情况下,可以从漫反射纹理图中查出取样,然后用得到的取样对漫反射颜色进行调制。有关更高级的应用将在本章后面部分讨论。

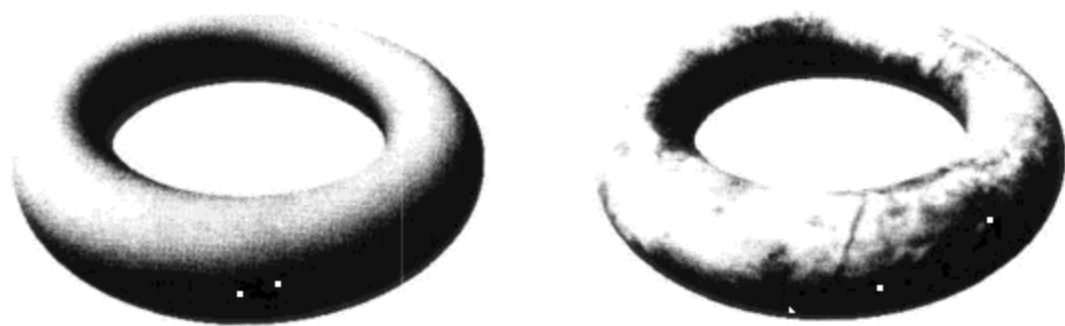


图 6.3 使用纹理图可以增加表面的细节

对于表面上的一点,用颜色 T 表示从纹理图中取得的对应于该点的过滤样本,用这个颜色来调制漫射颜色,就可以得到公式(6.6)的扩展版本

$$K_{\text{diffuse}} = DTA + DT \sum_{i=1}^n C_i \max\{N \cdot L_i, 0\} \quad (6.6)$$

从纹理图中取得的实际颜色取决于跟物体结合在一起的纹理坐标。纹理坐标要么预先计算,计算结果存储在三角网格的每个顶点中;要么在运行过程中计算,这样可以产生一些特殊的效果。对三角面进行渲染时,要使用公式(4.36)对纹理坐标进行插值。

每个顶点都可能都有 1~4 个纹理坐标,分别用 s, t, r 和 q 来标记。下面几节将介绍几种不同的纹理图,同时介绍在每种纹理图中如何使用纹理坐标查找纹理像素。

6.4.1 标准纹理图

在查找一维、二维和三维纹理图中的纹理元素时要分别用到一维、二维和三维纹理坐标。如图 6.4 所示,纹理图的宽度、高度和深度分别对应于 s, t, r 方向上的坐标值,取值区间为 0 到 1。

一维纹理图可以看作是在高度上只有一个像素的二维纹理图。同样,二维纹理图可以看作是在深度上只有一个像素的三维纹理图。当 t, r 坐标没有特别说明时,一般情况下可

以认为值都为 0。

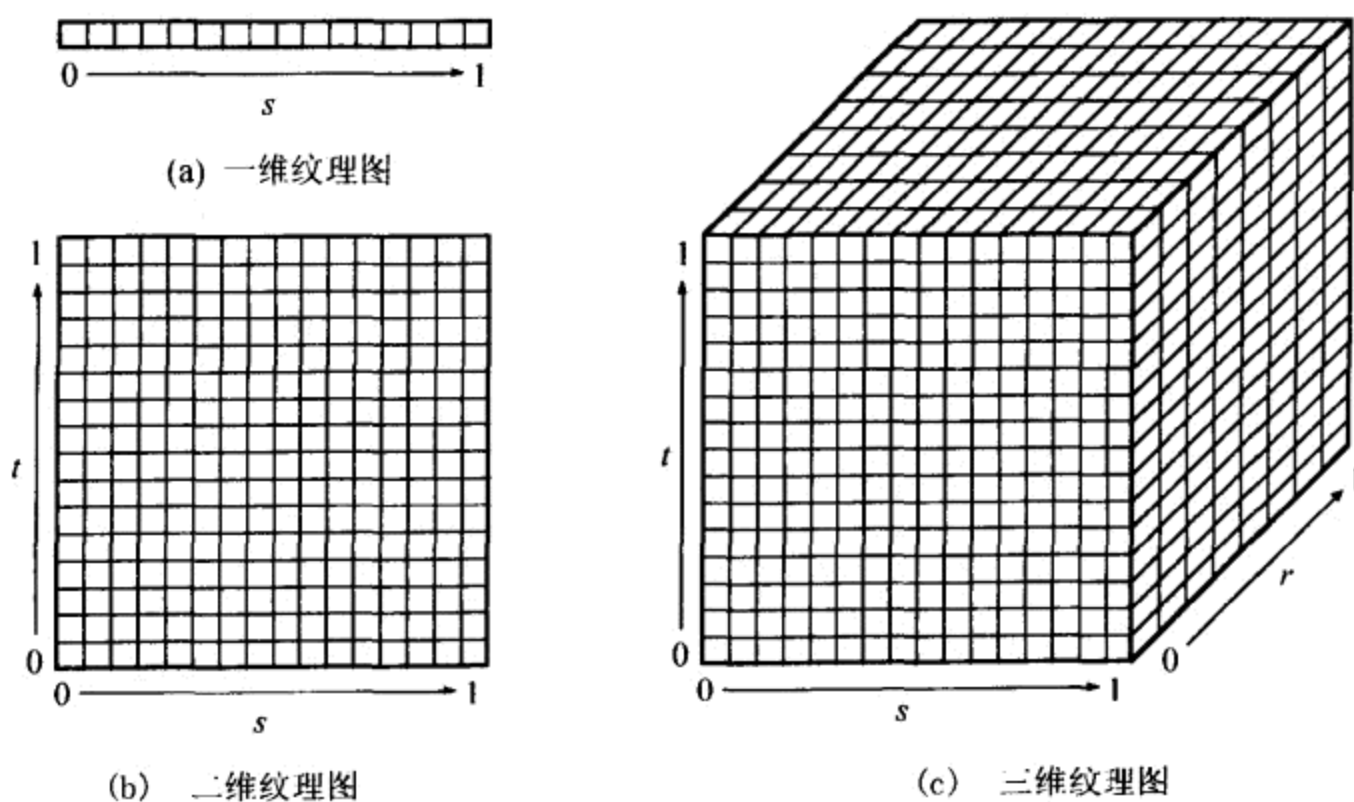


图 6.4 纹理空间

6.4.2 投影纹理图

第四个纹理坐标 q 用于投影纹理映射,在本节后面部分将介绍到它的具体应用。 q 坐标的作用与齐次坐标点中的 w 坐标非常类似,在没有特别说明的情况下,通常认为它的值为 1。插值坐标 s, t, r 要除以插值坐标 q 。对于端点纹理坐标为 $\langle s_1, t_1, r_1, q_1 \rangle$ 和 $\langle s_2, t_2, r_2, q_2 \rangle$ 的扫描线,可以用公式(4.36)来计算插值 s_3 和 q_3 ,其中要用到一个中间参数 u ($u \in [0, 1]$)。插值所得的 s_3 和 q_3 的商就是对纹理图进行取样的 s 坐标,为

$$s = \frac{s_3}{q_3} = \frac{(1-u) \frac{s_1}{z_1} + u \frac{s_2}{z_2}}{(1-u) \frac{q_1}{z_1} + u \frac{q_2}{z_2}} \quad (6.8)$$

使用类似的公式可以得到投影纹理坐标 t 和 r 。

使用投影纹理图可以模拟一种聚焦光源,该光源把一幅图像投影到场景中,这就是投影纹理图的一种应用。如图 6.5 所示,与聚焦光源的距离越远,得到的投影图像就越大。这样的效果可以通过以下方式得到:用一个 4×4 纹理矩阵来将物体的顶点位置坐标映射到纹理坐标 $\langle s, t, 0, q \rangle$,除以 q 之后得到校正的 2D 纹理坐标 $\langle s, t \rangle$,然后使得到的 2D 纹理坐标从投影图像中取样。

假设聚焦光源位于 P 点,方向为 U 。令单位向量 S 和 T 位于垂直于 U 的平面上,这样,可以使投影纹理图像的 s 轴和 t 轴与 S 和 T 的方向保持一致。对于被聚焦光照射的表面的每个点 $\langle x, y, z, 1 \rangle$,必须先变换到另一个坐标系中,在该坐标系中聚焦光源位于原点并且 x, y 和 z 轴分别与方向 S, T, U 保持一致,这个变换过程可以使用以向量 S, T, U, P 为列的矩阵的逆矩阵来完成。如果 S, T 是正交的(这时投影图像不会出现斜扭现象),则变

换矩阵为

$$M_1 = \begin{bmatrix} S_x & S_y & S_z & -\mathbf{S} \cdot \mathbf{P} \\ T_x & T_y & T_z & -\mathbf{T} \cdot \mathbf{P} \\ U_x & U_y & U_z & -\mathbf{U} \cdot \mathbf{P} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.9)$$

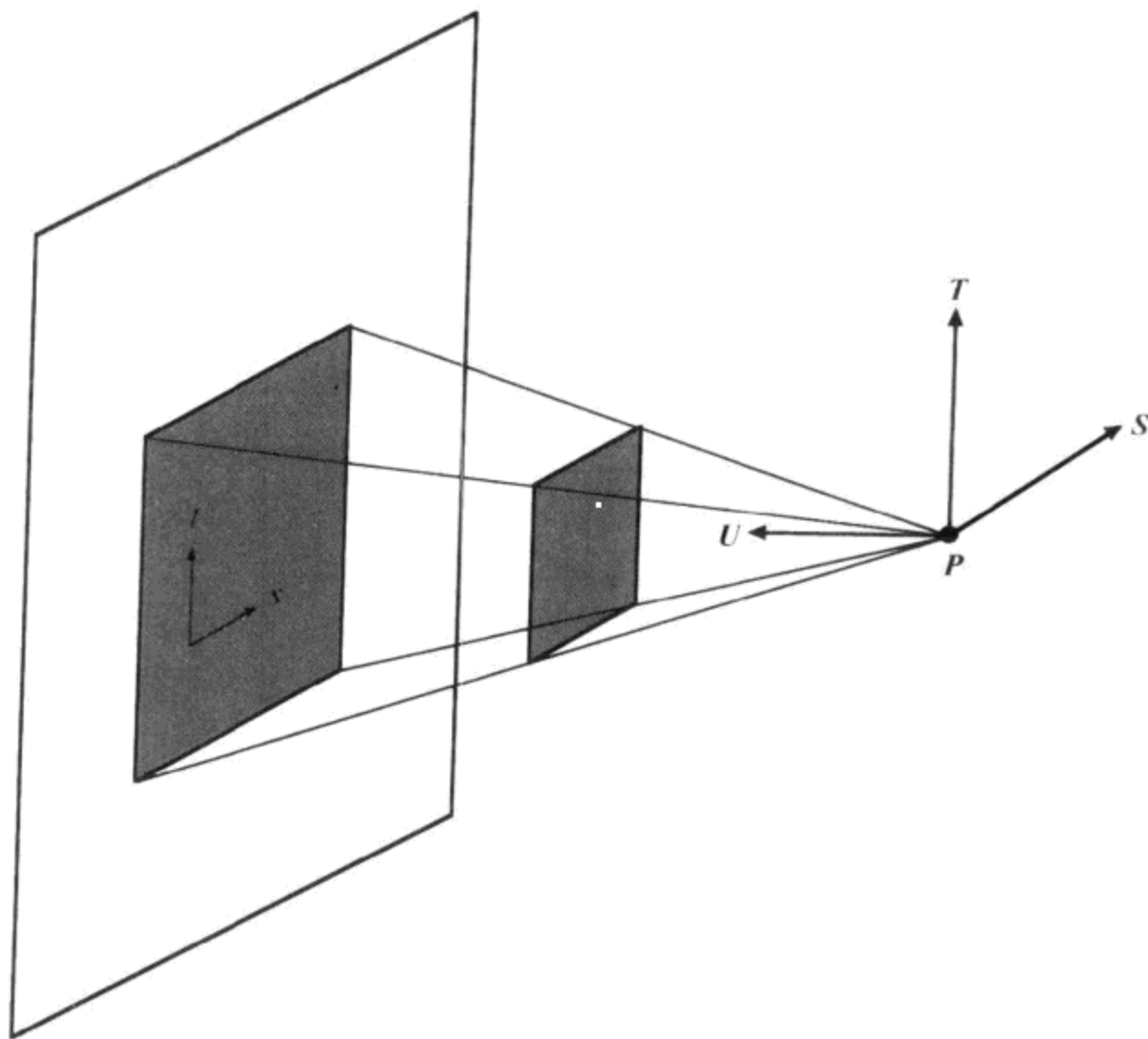


图 6.5 投影纹理图用于模拟将图像投影到场景中的聚焦光源

注意:因为 $\mathbf{S} \times \mathbf{T} = -\mathbf{U}$, 这个矩阵变换的结果为一个左手坐标系。

现在需要将公式(6.9)中的矩阵乘以一个实现投影的矩阵。就像定义视截体的焦距一样,可以用顶角 α 来定义聚焦光投影的焦距 e 为

$$e = \frac{1}{\tan(\alpha/2)} \quad (6.10)$$

设 a 为纹理图的高宽比,它的值等于纹理图的高除以宽。每个顶点位置都应被投影到距离聚焦光源为 e 的平面上,在这个平面上,要将 x 方向上的区间 $[-1, 1]$ 映射到 $[0, 1]$ 上, y 方向上的区间 $[-a, a]$ 映射到 $[0, 1]$ 上,实现这个过程的映射矩阵为

$$M_2 = \begin{bmatrix} e/2 & 0 & 1/2 & 0 \\ 0 & e/2a & 1/2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (6.11)$$

将映射结果中的 s 坐标和 t 坐标除以 q 坐标, 就会得到最终的投影结果。结合公式 (6.9) 和公式 (6.11) 中的矩阵, 根据公式 $M = M_2 M_1$, 就可以得到形成聚焦光图像投影效果的 4×4 纹理矩阵 M 。

6.4.3 立方体纹理图

立方体纹理图提供了对物体进行纹理映射的一种新方法, 常用立方体纹理图来近似地表示模型表面对周围环境的反射。如图 6.6 所示, 立方体纹理图由六个二维分量组成, 分别对应于立方体的六个面。 s, t 和 r 坐标组成了从立方体中心指向待采样纹理像素的方向向量。

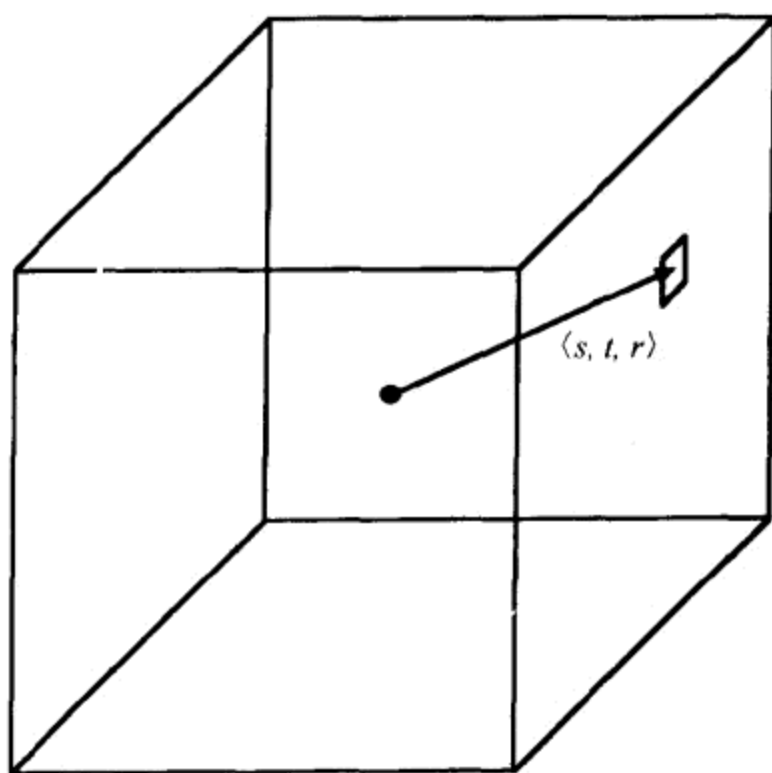


图 6.6 立方体纹理图由六个对应于立方体的面的分量组成

应该对哪个面进行采样是由绝对值最大的坐标的符号来决定的, 其他两个坐标除以最大坐标, 然后使用表 6.1 中的公式将这些坐标重新映射到区间 $[0, 1]$ 上, 以生成 2D 纹理坐标 $\langle s', t' \rangle$ 。这样在立方体纹理图相应的面上, 就可以使用相应的纹理坐标实现二维纹理图的采样。在图 6.7 中, 给出了在立方体各个面上的立方体纹理映射轴的方向。

表 6.1 用于计算 2D 坐标 $\langle s', t' \rangle$ 的公式

面	s'	t'
正 x	$\frac{1}{2} - \frac{r}{2s}$	$\frac{1}{2} + \frac{t}{2s}$
负 x	$\frac{1}{2} - \frac{r}{2s}$	$\frac{1}{2} + \frac{t}{2s}$
正 y	$\frac{1}{2} + \frac{s}{2t}$	$\frac{1}{2} + \frac{r}{2t}$
负 y	$\frac{1}{2} - \frac{s}{2t}$	$\frac{1}{2} + \frac{r}{2t}$

续表

面	s'	t'
正 z	$\frac{1}{2} + \frac{s}{2r}$	$\frac{1}{2} - \frac{t}{2r}$
负 z	$\frac{1}{2} + \frac{s}{2r}$	$\frac{1}{2} + \frac{t}{2r}$

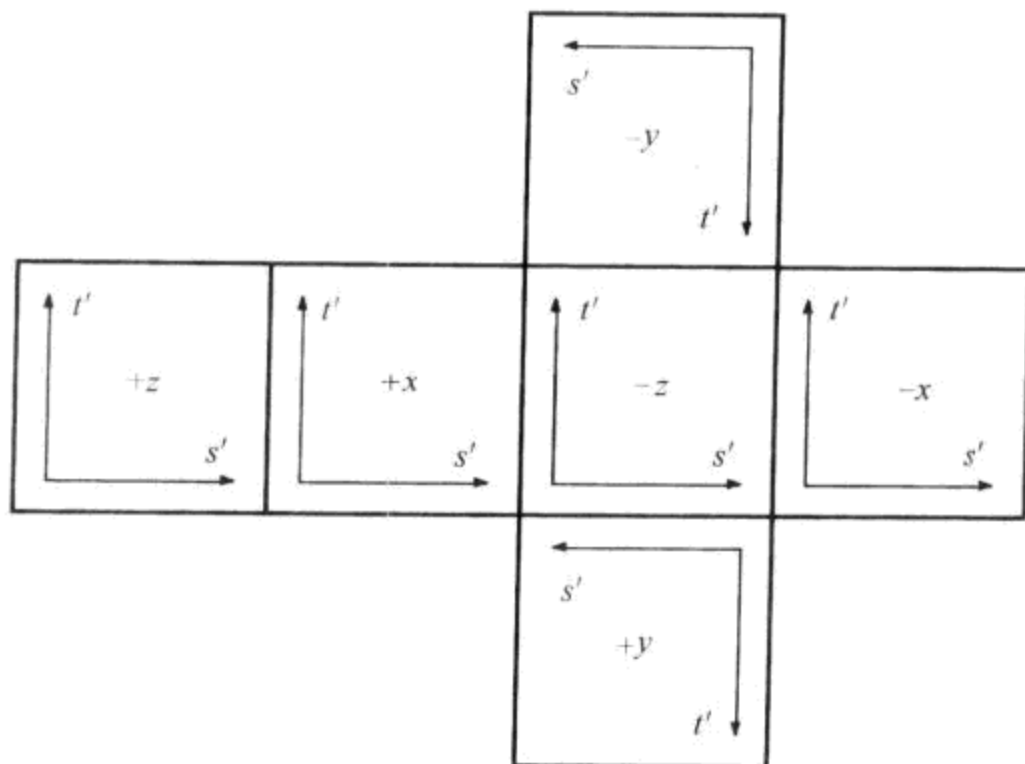


图 6.7 各个面上对应的立方体纹理映射轴的指向

与立方体纹理图一起使用的纹理坐标通常是在程序运行时产生的。例如,环境映射可以这样实现:计算出朝向摄像机方向的反射,然后将计算得到的反射信息存储在三角网格中每个顶点的 $\langle s, t, r \rangle$ 坐标中。反射方向的计算通常由计算机的硬件来完成的,所以具有很高的效率。

具有规格化向量的立方体纹理图是立方体纹理图的一种极具价值的应用。规格化立方体纹理图是这样一种立方体纹理图,在它的六个面上存储的不是图像颜色,而是向量阵列,每个向量以 RGB 的形式进行编码记录,计算如下公式:

$$\begin{aligned} \text{red} &= \frac{x+1}{2} \\ \text{green} &= \frac{y+1}{2} \\ \text{blue} &= \frac{z+1}{2} \end{aligned} \quad (6.12)$$

在立方体纹理图的面上,每个像素位置上实际存储的是像素采样的单位向量 $\langle s, t, r \rangle$ 。当进行像素级光照明计算时,就要用到规格化立方体纹理图,因为在整个三角面上进行曲面法向量插值时,所得到的法向量的长度恒小于 1。

6.5 镜面反射光

除了均匀的漫反射,表面还会形成强烈的光反射,反射的路线与入射方向和表面法线形

成的反射方向保持一致,其结果是在表面上出现耀眼的强光(高光),这就是镜面反射。与漫反射不同,镜面反射在表面上的可见性依赖于观察者所处的位置。

在图 6.8 中, N 是表面上 Q 点的法向量, V 为指向观察者的单位方向向量, L 为指向光源的单位方向向量, R 是根据公式(5.83)计算出来的对应于 L 的反射向量。当反射方向 R 指向观察者时,镜面强光最强;当 R 和指向观察者的向量 V 之间的角度增加时,强光的强度逐渐减弱。

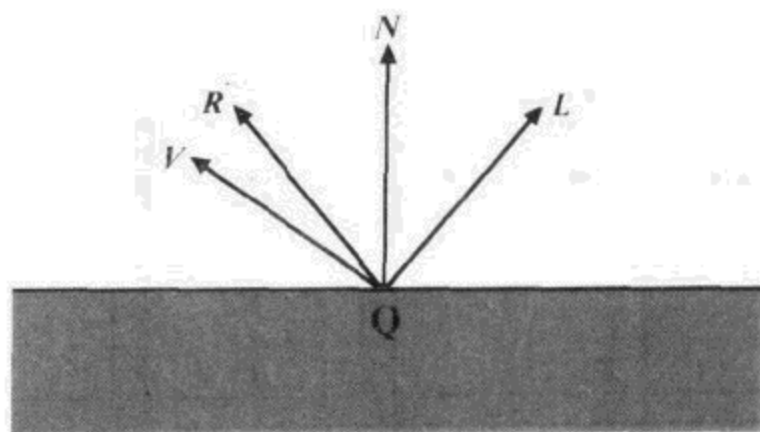


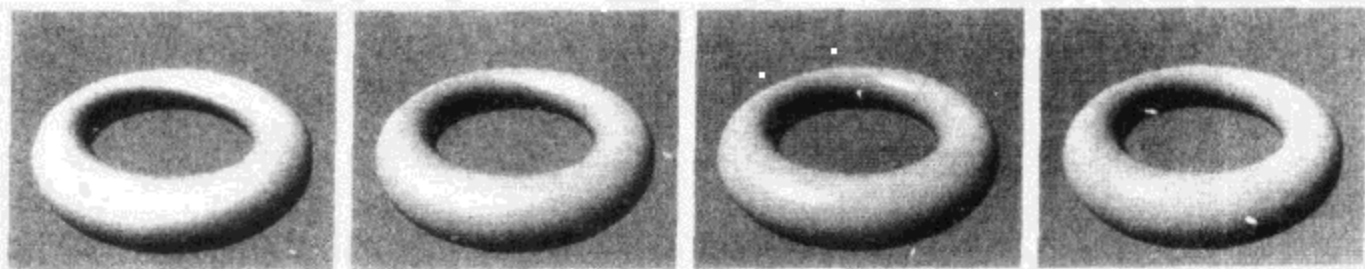
图 6.8 镜面反射的光强与 V 和 R 向量之间的夹角有关

这里给出一个对镜面强光进行描述的可信模型

$$SC \max |R \cdot V, 0|^m (N \cdot L > 0) \quad (6.13)$$

虽然几乎没有实际的物理基础与其对应,但该模型确实可信。用(6.13)式可以给出单一光源的镜面反射量,其中 S 是表面的镜面反射颜色, C 是入射光的光强, m 为镜面指数。 $(N \cdot L > 0)$ 表达式,当其为真时值为 1,否则值为 0,这样可以防止镜面强光在背向光源的平面上出现。

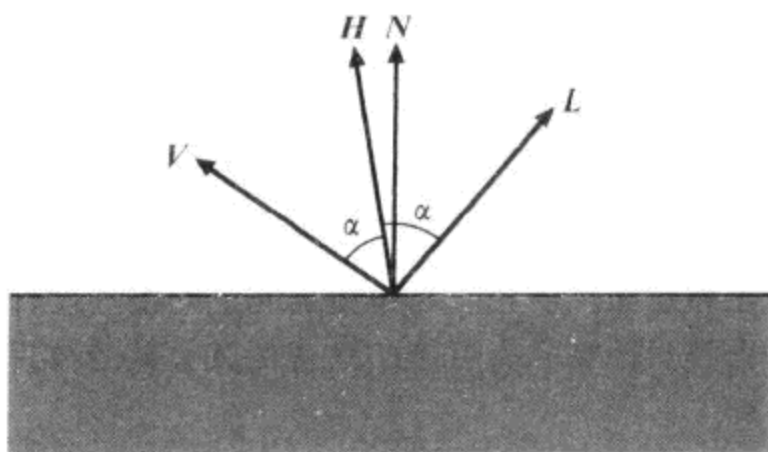
镜面指数 m 控制镜面强光的集中程度。如图 6.9 所示,小的 m 值产生模糊的强光,并在较大的范围内逐渐消失;大的 m 值产生明显的强光,并随着向量 V 和 R 之间的相互分离,很快地消退。



注:从左到右, m 的值依次为 2,10,50,100。

图 6.9 镜面指数 m 控制表面镜面强光的集中程度

下面介绍另外一种镜面强光计算公式,该公式在某些条件下需要较少的计算量,其中用到了一个叫做平分向量的向量。如图 6.10 所示,平分向量 H 刚好位于指向观察者的向量 V 和指向光源的向量 L 的中间,当平分向量 H 与法向量 N 指向相同时,镜面强光的强度最大。在这个模型中,可以用点积 $N \cdot H$ 代替公式(6.13)中的点 $R \cdot V$,虽然这会使镜面强光减弱的速率发生变化,但仍然不失原来模型的一般特性。

图 6.10 法向量 N 和平分向量 H 之间的夹角也可以用于确定镜面的强度

把 n 个光源相加,可以将光照明公式的镜面分量表示为

$$K_{\text{specular}} = S \sum_{i=1}^n C_i \max\{N \cdot H_i, 0\}^m (N \cdot L_i > 0) \quad (6.14)$$

其中, H_i 是第 i 个光源的平分向量,即

$$H_i = \frac{L_i + V}{\|L_i + V\|} \quad (6.15)$$

就像可以用纹理图对光照明公式的漫反射分量进行调制一样,也可以用一个纹理图对镜面分量进行调制,这样的纹理图称为光泽图,用它确定表面上每个点的镜面光强。用颜色 G 表示从光泽图得到的过滤样本,可以将镜面光照的公式扩展为

$$K_{\text{specular}} = SG \sum_{i=1}^n C_i \max\{N \cdot H_i, 0\}^m (N \cdot L_i > 0) \quad (6.16)$$

6.6 发射光

一些物体除了可以反射光,还可以发光。为了能够表示均匀发光的物体,在光照明公式中加入发射颜色 E 。也可以用一个发射映射图对发射颜色 E 进行调制,这里的发射映射图决定了表面上每个点发出的光的颜色和亮度。设颜色 M 是取自发射映射图的过滤样本,那么光照明公式的发射分量可以用下面的简单公式给出:

$$K_{\text{emission}} = EM \quad (6.17)$$

图 6.11 示例了发射映射图在圆环表面的应用。



图 6.11 对物体表面使用发射映射图可以确定表面的发光部分

6.7 明暗处理

在存储模型表面信息的时候,如表面上点的位置以及在这些点上的法向量,通常只存储三角网格的各个顶点的这些相关信息。正如在 4.4.2 节中讨论的那样,当渲染一个三角形时,要在整个三角面上对每个顶点已知的信息进行插值运算。传统的光照管道技术仅在网格的顶点处计算漫反射光和镜面反射光。当今更加先进的图形硬件使得对显示在屏幕上的每个单独像素进行整体光照明计算成为可能。结合任意数量的纹理图来确定三角表面的光亮度的方法称为明暗处理。

6.7.1 法向量的计算

对三角网格使用光照公式的时候,需要用到每个顶点的表面法向量。可以用叉积来计算一个三角形的法向量,一个顶点为 P_0, P_1 和 P_2 的三角形的单位法向量 N 为

$$N = \frac{(P_1 - P_0) \times (P_2 - P_0)}{\|(P_1 - P_0) \times (P_2 - P_0)\|} \quad (6.18)$$

这里假定当法向量指向观察者时,顶点的排列方式为逆时针,如图 6.12 所示。

计算一个顶点的法向量时,通常通过计算共有该点的所有三角形的法向量的平均值而求得。计算公式为

$$N_{\text{vertex}} = \frac{\sum_{i=1}^k N_i}{\|\sum_{i=1}^k N_i\|} \quad (6.19)$$

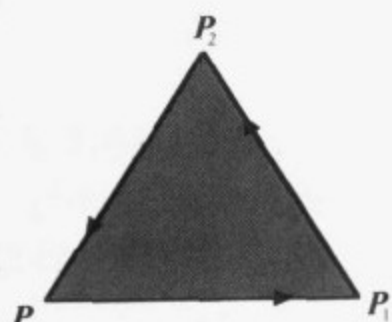


图 6.12 计算三角形的法向量

该公式用来计算 k 个三角形所共有的顶点的法向量 N_{vertex} ,每个三角形的法向量 N_i 对所得到的顶点法向量的作用是相等的。

图 6.13 给出了另外一种计算方法,它基于这样的事实:两个向量的叉积与这两个向量所形成的三角形的面积成正比。

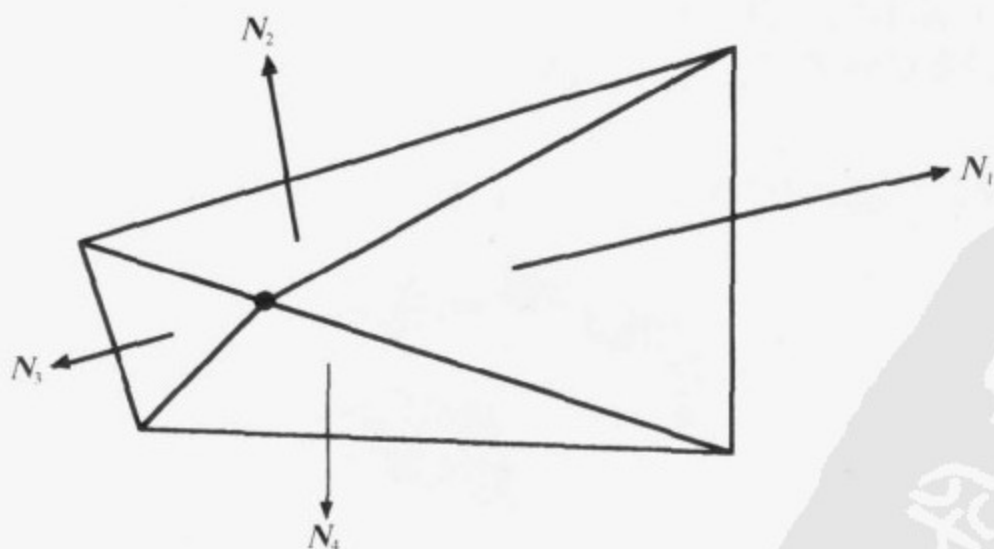


图 6.13 求共有同一顶点的所有三角形的法向量的平均值

在计算三角形的法向量的时候,我们不使用公式(6.18),而使用公式

$$\mathbf{N} = (\mathbf{P}_1 - \mathbf{P}_0) \times (\mathbf{P}_2 - \mathbf{P}_0) \quad (6.20)$$

该公式计算得到的是非规格化的三角形法向量,使用公式(6.19)求这些三角形的法向量的平均值,求得的结果就是所需的顶点法向量。三角形的面积越大,则对该顶点法向量的影响也就越大。对一些模型来说,这种方法所得到的顶点法向量更令人满意。

6.7.2 Gouraud 明暗处理

将计算得到的每个顶点的光亮度值在整个三角面上进行插值的过程称为 Gouraud 明暗处理。在能够计算像素级的光亮度的图形硬件出现之前,漫反射和镜面反射的颜色只能用在三角网格的各个顶点上计算。Gouraud 明暗处理法使用下面的公式计算每个顶点的颜色:

$$\mathbf{K}_{\text{primary}} = \mathbf{E} + \mathbf{DA} + \mathbf{D} \sum_{i=1}^n C_i \max\{\mathbf{N} \cdot \mathbf{L}_i, 0\} \quad (6.21)$$

$$\mathbf{K}_{\text{secondary}} = \mathbf{S} \sum_{i=1}^n C_i \max\{\mathbf{N} \cdot \mathbf{H}_i, 0\}^m (\mathbf{N} \cdot \mathbf{L}_i > 0)$$

对计算得到的顶点颜色在三角面上插值,每个像素的颜色 \mathbf{K} 的计算公式为

$$\mathbf{K} = \mathbf{K}_{\text{primary}} \circ \mathbf{T}_1 \circ \mathbf{T}_2 \circ \cdots \circ \mathbf{T}_k + \mathbf{K}_{\text{secondary}} \quad (6.22)$$

其中 \mathbf{T}_k 表示从 k 个纹理图中的一个纹理图中取样得到的颜色,运算符“ \circ ”是几种可能的纹理组合操作符之一,包括调制和加法。

6.7.3 Phong 明暗处理

用 Phong 模型对三角面进行明暗处理时,插值的对象不是各个顶点的光亮度值,而是各个顶点的法向量,根据法向量插值来计算各个像素的光亮度。对于能够进行像素级复杂运算(该过程称为像素明暗处理或分段明暗处理)的图形硬件,可以将其配置为按照下面的公式对三角面上的各个像素进行计算:

$$\begin{aligned} \mathbf{K} &= \mathbf{K}_{\text{emission}} + \mathbf{K}_{\text{diffuse}} + \mathbf{K}_{\text{specular}} \\ &= \mathbf{EM} + \mathbf{DAT} + \sum_{i=1}^n C_i [\mathbf{DT}(\mathbf{N} \cdot \mathbf{L}_i) + \mathbf{SG}(\mathbf{N} \cdot \mathbf{H}_i)^m (\mathbf{N} \cdot \mathbf{L}_i > 0)] \quad (6.23) \end{aligned}$$

为了简化,在此省略了最大值函数。应该注意的是,公式中将漫反射和镜面反射的点积设为零值。 n 个光源中各个光源的光强 C_i 仍然在各个顶点处计算,仍然要在三角面上进行插值。计算所得的光强和插值所得的法向量可用于计算各个像素点的 \mathbf{K} 值。当然,不要求公式(6.23)的每个分量都存在。

Phong 明暗模型相比 Gouraud 明暗模型的突出优点是,它对镜面反射的建模能力远远强于 Gouraud 明暗处理模型,这是因为 $\mathbf{N} \cdot \mathbf{H}$ 的计算是在每个像素上进行的。当一束集中的镜面强光照到三角形内部时,Gouraud 明暗处理所得到的结果不尽人意,因为在三角形顶点处计算得到的镜面反射分量并不能表示三角面上其他点处的实际值。

当使用 Phong 明暗模型的时候,会出现一个问题:插值所得的法向量不能像在顶点时那样保持单位长度。在稠密的棋盘格子模型中,相邻顶点的法向量在方向上的区别很小,因此不会产生视觉上难以接受的人工噪声,但是在大多数模型中,会在每个三角形内部明显地

出现镜面光变得黯淡的现象。利用规格化的立方体映射图可解决这个问题(参见 6.4.3 节)。将法向量传给纹理引擎作为 $\langle s, t, r \rangle$ 映射坐标,得到的输出结果是 RGB 颜色编码格式的单位向量。

6.8 凹凸映射

当物体从任何非边缘的方向观察的时候,观察者见到的表面细节通常是由表面的光照方式决定的。渲染过程中,每个像素的光照度是由光照计算公式所用的法向量决定的。到目前为止,只是先计算三角网格顶点的法向量,然后在其他地方使用平滑插值向量。这种粗糙的解决方法难以解决比网格中典型三角形还小的光照细节问题。相对于此,凹凸映射技术能够解决这样的问题,它通过使用纹理映射在每个像素上对法向量进行扰动,从而为观察者提供更详细的光照细节。

6.8.1 凹凸映射图的构造

对法向量进行扰动的高分辨率信息存储在一种三维向量(称为块图或法线图)的二维数组中,这里的三维向量称为凹凸映射图或法向量映射图。凹凸映射图中的每个向量表示一个方向,在这个方向上,法向量相对于插值所得法向量做相对偏向。向量 $\langle 0, 0, 1 \rangle$ 表示一个没被扰动的法向量,任何其他的向量则表示对影响光亮度公式的向量的修改。

从一个高度图中提取向量,就可以构造一个凹凸图,高度图中的内容表示的是平面上每个像素的平面高度。为了从高度图中得到某一像素所对应的法向量,首先要计算 s 和 t 方向上的切向量,这里的切向量依赖于邻近两像素间的高度差。在具有 $w \times b$ 个像素的高度图中,用 $H(i, j)$ 表示存储在 $\langle i, j \rangle$ 位置的高度值,这样可以将 s 和 t 方向上的切向量 $S(i, j)$ 和 $T(i, j)$ 表示为

$$\begin{aligned} S(i, j) &= \langle 1, 0, aH(i+1, j) - aH(i-1, j) \rangle \\ T(i, j) &= \langle 0, 1, aH(i, j+1) - aH(i, j-1) \rangle \end{aligned} \quad (6.24)$$

公式中的常量 a 是用于调节高度值范围的比例系数,用它可以控制对法向量扰动的程度。如果用 S_z 和 T_z 分别表示 $S(i, j)$ 和 $T(i, j)$ 的 z 分量,则法向量 $N(i, j)$ 可通过下面的叉积得到:

$$N(i, j) = \frac{S(i, j) \times T(i, j)}{\|S(i, j) \times T(i, j)\|} = \frac{\langle -S_z, -T_z, 1 \rangle}{\sqrt{S_z^2 + T_z^2 + 1}} \quad (6.25)$$

根据公式(6.12),法向量的分量用 RGB 颜色编码方式进行记录。在图 6.14 中给出了一个灰度高度图,以及用公式(6.25)得到的对应的凹凸映射图。凹凸映射图中包含了被扰动的法向量。图中大部分都是淡紫色,这是因为对于未被扰动的法向量 $\langle 0, 0, 1 \rangle$, 计算得到的 RGB 颜色为 $\langle \frac{1}{2}, \frac{1}{2}, 1 \rangle$ 。

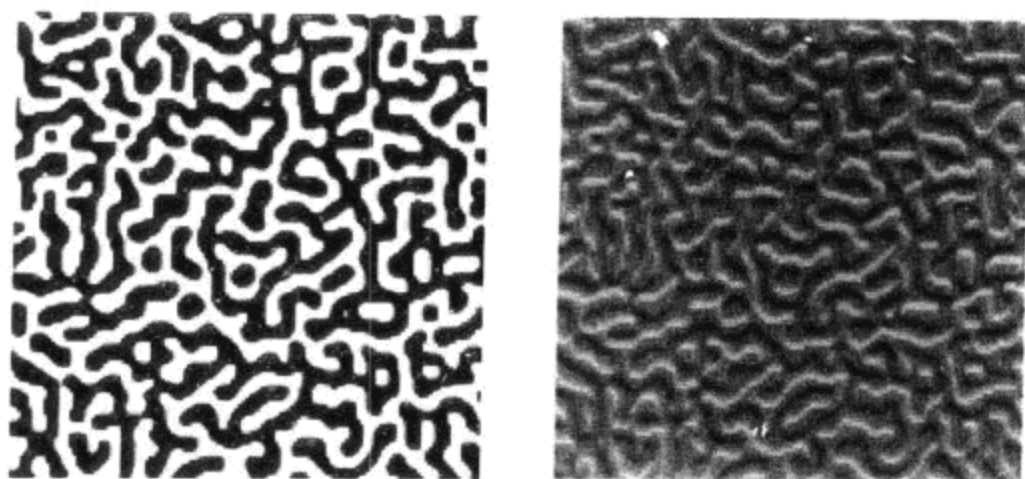


图 6.14 高度图和相应的凹凸映射图

6.8.2 切线空间

因为向量 $\langle 0, 0, 1 \rangle$ 在凹凸映射中表示未经扰动的法向量, 需要将它与光亮度公式中经常使用的插值法向量相对应。这可以通过在每个顶点位置构造一个坐标系来实现, 在该坐标系中顶点法向量总是指向 z 轴正方向。除了法向量, 在每个顶点位置还需要两个与表面相切的向量, 这两个向量形成正交基。这样所形成的坐标系统称为切线空间或顶点空间, 如图 6.15 所示。在顶点位置, 切线空间与顶点切平面和法向量相对应。

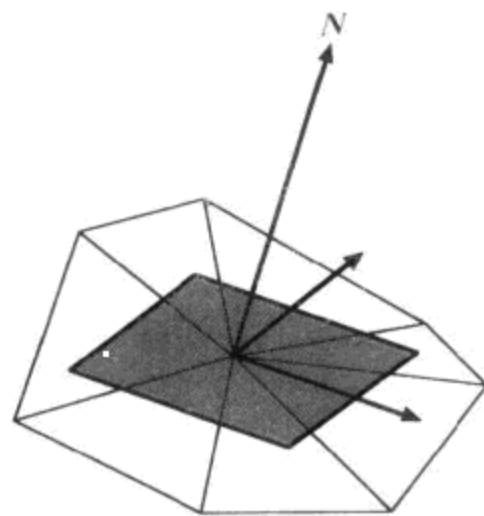


图 6.15 切线空间

一旦建立了三角网格的每个顶点的切线空间坐标系, 就可以在每个顶点位置计算指向光的方向向量 L , 并将 L 变换到切线空间, 然后对切线空间中的向量 L 在整个三角面上进行插值运算。由于切线空间里的向量 $\langle 0, 0, 1 \rangle$ 对应的是法向量, 所以切线空间中指向光的方向向量 L 与凹凸映射图中的一个取样的点积就可以产生有效的 Lambertian 反射。

必须选择每个顶点的切向量, 以使它们与凹凸映射图的纹理空间保持一致。对于参数函数所生成的曲面, 通常只需简单地对每个参数求导就可以得到切线。但是, 由于任意三角网格可能在任何方向上使用凹凸映射, 所以需要有一种更加通用的计算方法来计算每个顶点的切向量。

6.8.3 切向量的计算

我们的目标是在每个顶点处找到一个 3×3 矩阵, 使用该矩阵可以将向量从对象空间转换到切线空间。为了达到这个目的, 先考虑比较直观的问题, 就是在相反的方向上如何将向量从切线空间转换到对象空间。因为顶点的法向量与切线空间中 $\langle 0, 0, 1 \rangle$ 相对应, 所以可以知道切线空间中的 z 轴总是映射为顶点的法向量。

我们希望切线空间是这样安排的: x 轴与凹凸映射图的 s 方向保持一致, y 轴与凹凸映射图的 t 方向一致。这样, 如果 P 表示三角形内部一点, 可以给出这样的表达式

$$P - E = (s - s_E)T + (t - t_E)B \quad (6.26)$$

其中 T 和 B 是与纹理映射图保持一致的切向量(字母 B 代表副法线), E 是三角形的

一个顶点的位置, $\langle s_E, t_E \rangle$ 是该顶点的纹理坐标。

假设三角形顶点位置分别为 E, F 和 G , 并且它们所对应的纹理坐标分别为 $\langle s_E, t_E \rangle$, $\langle s_F, t_F \rangle$ 和 $\langle s_G, t_G \rangle$ 。以顶点 E 作为相对基准, 可以大大简化计算过程, 所以设

$$\begin{aligned} P &= F - E \\ Q &= G - E \end{aligned} \quad (6.27)$$

并且设

$$\begin{aligned} \langle s_1, t_1 \rangle &= \langle s_F - s_E, t_F - t_E \rangle \\ \langle s_2, t_2 \rangle &= \langle s_G - s_E, t_G - t_E \rangle \end{aligned} \quad (6.28)$$

我们所要解的就是下面关于 T 和 B 的方程:

$$\begin{aligned} P &= s_1 T - t_1 B \\ Q &= s_2 T - t_2 B \end{aligned} \quad (6.29)$$

这是一个线性系统, 有 6 个未知变量(对应于 T 和 B 的各有 3 个)和 6 个等式(两个等式中的 x, y 和 z 分量)。可以将上式写成矩阵形式

$$\begin{bmatrix} P_x & P_y & P_z \\ Q_x & Q_y & Q_z \end{bmatrix} = \begin{bmatrix} s_1 & t_1 \\ s_2 & t_2 \end{bmatrix} \begin{bmatrix} T_x & T_y & T_z \\ B_x & B_y & B_z \end{bmatrix} \quad (6.30)$$

两边都乘以 $\langle s, t \rangle$ 矩阵的逆, 可以得到

$$\begin{bmatrix} T_x & T_y & T_z \\ B_x & B_y & B_z \end{bmatrix} = \frac{1}{s_1 t_2 - s_2 t_1} \begin{bmatrix} t_2 & -t_1 \\ -s_2 & s_1 \end{bmatrix} \begin{bmatrix} P_x & P_y & P_z \\ Q_x & Q_y & Q_z \end{bmatrix} \quad (6.31)$$

这样, 可以得到顶点为 E, F 和 G 的三角形的未规格化的切向量 T 和 B 。为了得到某一顶点的切向量, 需求共有该顶点的三角形的切向量的平均值, 这跟计算顶点法向量的方法很类似。在相邻的三角形有不连续的纹理映射(见图 6.16)的情况下, 边界上的顶点上通常存在着不同的复制顶点, 它们各有不同的映射坐标, 此时就不需要求这样的三角形的平均值, 因为求平均值所得的结果将不能准确地表示两个三角形各自在凹凸映射图上的方向。

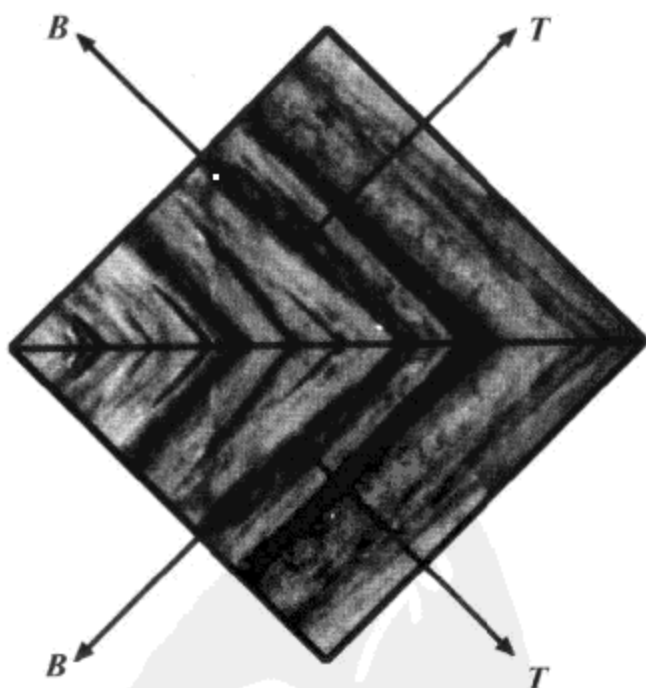


图 6.16 相邻的三角形可能有不同的纹理图方向

一旦得到了顶点的法向量 \mathbf{N} 和切向量 \mathbf{T} 与 \mathbf{B} , 利用下面的矩阵, 就可以实现从切线空间到对象空间的变换

$$\begin{bmatrix} T_x & B_x & N_x \\ T_y & B_y & N_y \\ T_z & B_z & N_z \end{bmatrix} \quad (6.32)$$

在实现相反方向的变换时(从对象空间到切线空间——要对指向光的方向向量做这样的变换), 只需简单地使用该矩阵的逆矩阵。因为切向量不必相互垂直, 也不必与法向量垂直, 所以通常情况下该矩阵的逆矩阵通常不等于它的转置矩阵。然而, 完全可以假设三个向量至少是接近于正交的, 所以使用 Gram-Schmidt 算法(见算法 1.16)进行正交规格化不会引起任何不可接受的扭曲。在正交规格化的过程中, 新的(仍未规格化的)切向量 \mathbf{T}' 和 \mathbf{B}' 可以表示为

$$\begin{aligned} \mathbf{T}' &= \mathbf{T} - (\mathbf{N} \cdot \mathbf{T})\mathbf{N} \\ \mathbf{B}' &= \mathbf{B} - (\mathbf{N} \cdot \mathbf{B})\mathbf{N} - (\mathbf{T}' \cdot \mathbf{B})\mathbf{T}' \end{aligned} \quad (6.33)$$

对这些向量进行规格化, 并将它们作为顶点的切线和副法线进行存储, 这样可以使下面的矩阵:

$$\begin{bmatrix} T'_x & T'_y & T'_z \\ B'_x & B'_y & B'_z \\ N_x & N_y & N_z \end{bmatrix} \quad (6.34)$$

将指向光的方向向量从对象空间变换到切线空间。求变换后的指向光的方向向量与凹凸映射图一个取样之间的点积, 就可以得到正确的 Lambertian 漫反射光照值。

不需要额外的数组来存储每个顶点的副法线, 因为从叉积 $\mathbf{N} \times \mathbf{T}'$ 就可以得到 $m\mathbf{B}'$, 其中的 $m = \pm 1$ 给出了切线空间的手向性(左手坐标系或右手坐标系)。由于从 $\mathbf{N} \times \mathbf{T}'$ 中获得的副法线 \mathbf{B}' 可能会指向错误的方向, 因此必须存储每个顶点的手向值, 表示手向性的等于公式(6.34)中矩阵的行列式值。也许有人会发现, 将每个顶点的切向量 \mathbf{T}' 以四维向量形式存储显得非常方便, 其中四维向量的 w 坐标值等于 m 。这样, 副法线 \mathbf{B}' 的计算公式为

$$\mathbf{B}' = T'_w(\mathbf{N} \times \mathbf{T}') \quad (6.35)$$

计算叉积时忽略了 w 坐标。这对顶点编程很有益处, 因为这样可以避免用额外的数组来保存每个顶点的 m 值。

6.8.4 实现

凹凸映射运算可以分为对每个顶点的计算和对每个像素的计算两部分。在每个顶点位置, 必须计算出指向光的方向向量 \mathbf{L} 和平分向量 \mathbf{H} , 并使用公式(6.34)将它们变换到切线空间。程序 6.1 给出了一个顶点程序, 实现了用一个定向光源(\mathbf{L} 是常数)照射表面的计算。

【程序 6.1】 这个顶点程序实现了凹凸映射所需的基本计算。标准正交向量 \mathbf{T}' 从顶点的属性数组 6 读取, 副法线 \mathbf{B}' 用公式(6.35)计算得到。切线空间的指向光源的方向向量 \mathbf{L} 存储在主颜色中, 切线空间的平分向量存储在次颜色中。凹凸映射图的纹理单元为 0, 而普通纹理图的纹理单元为 1。

!VP1.0

```

#   c[0-3]= 投影 * 模型图  矩阵
#   c[4]= 常量 (0.5,0.5,0.5,0.0)
#   c[5]= 物体空间中摄像机的位置 position
#   c[6]= 物体空间中光的法向 L

DP4   o[HPOS].x,c[0],v[OPOS];           # 变换顶点
DP4   o[HPOS].y,c[1],v[OPOS];
DP4   o[HPOS].z,c[2],v[OPOS];
DP4   o[HPOS].w,c[3],v[OPOS];

MOV   R0,v[NRML];                       # 装入法向量 N
MOV   R1,v[6];                           # 装入切向量 T

MUL   R2.xyz,R0.zxyw,R1.yzxw;           # B=N×T
MAD   R2.xyz,R0.yzxw,R1.zxyw,-R2;
MUL   R2,R2,R1.w;                       # B*=T.w

ADD   R3,c[5],-v[OPOS];                 # R3= 视线方向 V
DP3   R5,R3,R3;                          # 规格化 T
RSQ   R5.x,R5.x;
MUL   R3.xyz,R3,R5.x;

ADD   R4,R3,c[6];                       # R4= 平分向量 H
DP3   R6,R4,R4;                          # 规格化 H
RSQ   R6.x,R6.x;
MUL   R4.xyz,R4,R6.x;

DP3   R7.x,R1,c[6];                      # 将 L 变换到切线空间
DP3   R7.y,R2,c[6];
DP3   R7.z,R0,c[6];
MAD   o[COL0],R7,c[4],c[4];             # 编码为 RGB 格式

DP3   R8.x,R1,R4;                        # 将 L 变换到切线空间
DP3   R8.y,R2,R4;
DP3   R8.z,R0,R4;
MAD   o[COL1],R8,c[4],c[4];             # 编码为 RGB 格式

```

```

MOV    o[TEX0],v[TEX0];          # 复制凹凸映射坐标
MOV    o[TEX1],v[TEX1];          # 复制纹理映射坐标
END

```

必须计算每个像素的点积 $N \cdot L$ 和 $N \cdot H$, 其中法向量 N 是从凹凸映射图中取样得到的, 向量 L 和 H 是根据每个顶点的插值所得。因为这些是插值向量, 所以它们的数量模有所变小, 这使得三角形的内部比实际效果要显得黯淡些。这种效果一般情况下并不明显, 但对于缺乏足够网格的模型却需要用到规格化的立方体映射图。对于有足够纹理单元的硬件, 可以修改程序 6.1, 以使切线空间的向量 L 和 H 可以在立方体映射图中进行索引, 这里的立方体映射图可以高效地规格化纹理坐标。为了达到这个要求, 将规格化的立方体映射图设为纹理单元 2 和纹理单元 3, 然后用下面的代码代替变换 L 和 H 的代码部分。只拥有两个纹理单元的硬件则要求使用多通道技术进行渲染。

```

DP3    o[TEX2].x,R1,c[6];        # 将 L 变换到切线空间
DP3    o[TEX2].y,R2,c[6];        # <s,t,r> = L
DP3    o[TEX2].z,R0,c[6];

DP3    o[TEX3].x,R1,R4;          # 将 H 变换到切线空间
DP3    o[TEX3].y,R2,R4;          # <s,t,r> = H
DP3    o[TEX3].z,R0,R4;

```

每个像素的点积可用 GL-EXT-texture-env-dot3 扩展功能或 GL-NV-register-combiners 扩展功能来计算。通过连续平方可使 $N \cdot H$ 值的幂增至 1, 但这只允许用于小的二次幂指数的情况。OpenGL 实现了相关纹理读取功能(比如使用 GL-NV-texture-shader 扩展功能), 使得使用任意的镜面反射分量成为可能, 这时需要将 $S(N \cdot H)^m$ 的值存储到一个一维纹理映射图中。

6.9 物理反射模型

在此之前所介绍的表面光反射计算方法, 在大多数情况下计算开销不高, 而且在许多情况下产生的视觉效果令人满意, 但这些方法采用的不是精确的模型, 不能精确地反映反射光的物理分布。为了获得更高的真实性, 需要使用更好的表面微观结构模型, 同时还要用到少许的电磁理论。

6.9.1 双向反射分布函数

当光束照射到曲面表面时, 光束所包含的辐射能量会重新分布, 我们的目标就是对这种重新分布的方式进行建模。光束的一部分能量会被表面吸收, 可能有另外一部分能量穿过

表面继续传递,剩下的能量则都被反射出去。反射的能量通常分散到各个方向上,没有统一的形式。我们要介绍的双向反射分布函数(BRDF)以指向光源的方向向量 L 和反射方向向量 R 为参数,返回值为在反射方向 R 上对来自 L 的人射光的反射量。

为了精确地定义 BRDF,首先要介绍辐射度学(研究通过辐射进行能量传输的学科)领域中的一些术语。光源发出或表面接收到的辐射功率(单位时间的能量)称为通量,以瓦特作为测量单位。单位面积的光源发出的或单位面积表面接收到的辐射功率称为通量密度,以瓦特每平方米($W \cdot m^{-2}$)作为测量单位。由表面发出的通量密度称为表面的辐射度,而入射到表面的通量密度称为辐照度。

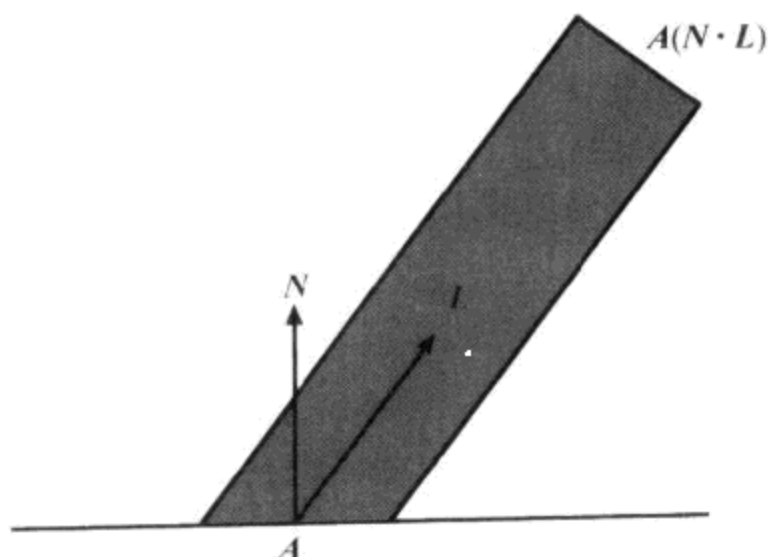


图 6.17 表面上面积为 A 的区域的人射通量密度

图 6.17 示例了一个光源向一个面积为 A 的表面发射 P 瓦特能量的情况。表面接收到的功率等于光源发射出的功率,但是由于 Lambertian 效应,吸收和发射的通量密度是不等的。光束的面积等于 $A(N \cdot L)$,其中 N 是表面的单位法向量, L 是指向光源的单位方向向量。则由光源发出的通量密度 Φ_E 为

$$\Phi_E = \frac{P}{A(N \cdot L)} \quad (6.36)$$

因为表面上的人射通量密度 Φ_I 等于 P/A ,所以有下面的关系式:

$$\Phi_I = \Phi_E(N \cdot L) \quad (6.37)$$

光照射到表面的方向是用立体角来衡量的,立体角的定义类似于平面角的定义,只不过它的定义是在三维中进行的。如图 6.18 所示,平面角 θ 的弧度等于在圆上所扫过的弧长 l 除以圆的半径,即 $\theta = l/r$ 。将这个公式扩展到三维时,在半径为 r 的球面上面积为 A 的区域所对应的立体角 $\omega = A/r^2$,即立体角等于立体角所对应的表面面积除以球面半径的平方。立体角的度量单位是立体弧度,缩写为 sr。由于半径为 r 的球面的表面积等于 $4\pi r^2$,则整个球面的立体角的立体弧度为 4π 。

立体角的微分 $d\omega$ 可以用极角的微分 $d\theta$ 和方位角的微分 $d\phi$ 来表示。如图 6.19 所示,极角 θ 处的圆平行于 xy 平面,通过点 (r, θ, ϕ) ,半径为 $r \sin \theta$ 。因此在该圆上,方位角方向上的弧长的微分等于 $r \sin \theta d\phi$,将该值乘以极角方向上的弧长的微分 $r d\theta$,就可以得到表面积微分 dA 的计算公式

$$dA = r^2 \sin \theta d\theta d\phi \quad (6.38)$$

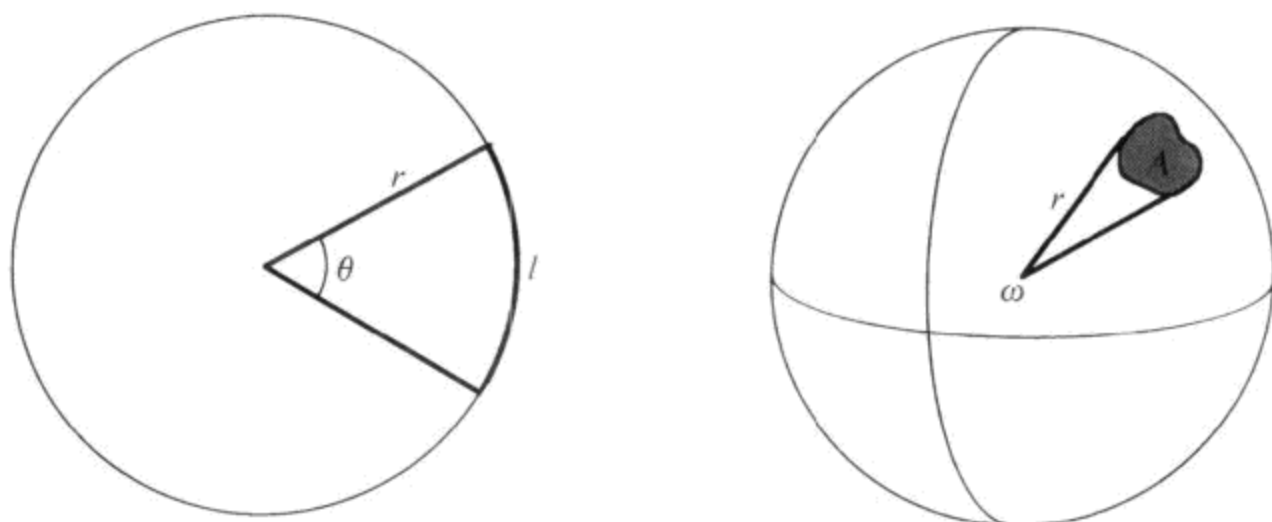


图 6.18 平面角和立体角

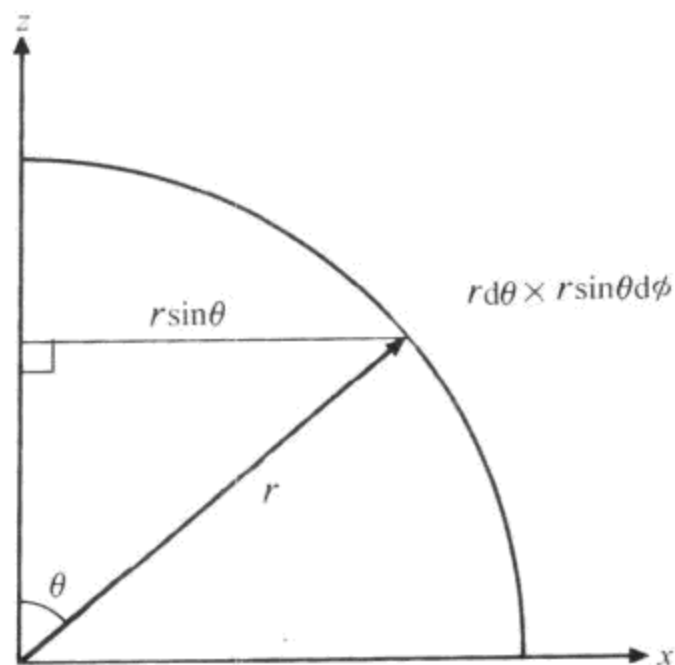


图 6.19 表面积的微分

公式两边除以 r^2 后, 就可以得到对应的立体角的微分 $d\omega$

$$d\omega = \sin\theta d\theta d\phi \quad (6.39)$$

辐射强度是用来描绘每单位立体角的辐射通量密度的术语, 以瓦特每平方米每立体弧度 $\text{W}\cdot\text{m}^{-2}\cdot\text{sr}^{-1}$ 作为度量单位。

微分面积 dA 所接收到的辐照度(通量密度) Φ_I 等于下面公式给出的关于 $C_I(\mathbf{L})$ 的积分(角 θ 和 ϕ 是对应于方向 \mathbf{L} 的极角和方位角):

$$\Phi_R = \int_{\Omega} C_I(\mathbf{L}) d\omega = \int_0^{2\pi} \int_0^{\pi/2} C_I(\theta, \phi) d\phi \quad (6.40)$$

其中, $C_I(\mathbf{L})$ 为接收到的辐射强度, 指向光的方向向量 \mathbf{L} 在表面上的单位半球面区域 Ω 内变化。

基于同样的原因, 表面吸收的通量密度与光源发射的通量密度之间的关系可用公式(6.37)给出, 表面吸收的辐射强度 C_I 和光源发射的辐射强度 C_E 之间的关系为

$$C_I = C_E(\mathbf{N} \cdot \mathbf{L}) = C_E \cos\theta \quad (6.41)$$

因此,可以将公式(6.40)改写为

$$\Phi_I = \int_{\Omega} C_E(\mathbf{L})(\mathbf{N} \cdot \mathbf{L})d\omega = \int_0^{2\pi} \int_0^{\pi/2} C_E(\theta, \phi) \cos\theta d\theta d\phi \quad (6.42)$$

表面上一点处的双向反射率 $\rho(\mathbf{V}, \mathbf{L})$ 是关于指向观察者的方向向量 \mathbf{V} 和指向光的方向向量 \mathbf{L} 的函数,它等于反射辐射强度的微分与入射辐射强度的微分之比

$$\rho(\mathbf{V}, \mathbf{L}) = \frac{dC_R}{d\Phi_I} = \frac{dC_R}{C_E(\mathbf{L})(\mathbf{N} \cdot \mathbf{L})d\omega} \quad (6.43)$$

函数 $\rho(\mathbf{V}, \mathbf{L})$ 是 BRDF, 在下面的公式中使用该函数可以计算出表面在某一方向上的反射光的辐射强度

$$DC_R = \rho(\mathbf{V}, \mathbf{L})C_E(\mathbf{L})(\mathbf{N} \cdot \mathbf{L})d\omega \quad (6.44)$$

定向光源、点光源以及聚焦光源都是在单一的方向上照射表面上的点的,因此,计算在指向观察者的方向上反射 n 个光源的 $C_R(\mathbf{V})$ 值的时候,不使用积分公式(6.44),而只是简单地在指向光源的离散的方向向量 \mathbf{L}_i 上求和。

$$C_R(\mathbf{V}) = \sum_{i=1}^n \rho(\mathbf{V}, \mathbf{L}_i)C_i(\mathbf{N} \cdot \mathbf{L}_i) \quad (6.45)$$

到目前为止,在讨论 BRDF 的时候还没有提及有关颜色的问题。除了是入射和反射的方向的函数之外,BRDF 也应该是光的波长函数。在需要能够覆盖整个光谱的精确反射模型的具体应用中,典型的做法是求几种波长的 BRDF,然后把所得的值拟合成一条曲线。在实时计算机图形系统中发现,把 BRDF 看作以入射光的 RGB 值为参数、返回值为反射光的 RGB 值的函数,是能够满足需要的。从现在开始,假设涉及到 BRDF 的所有运算,都是针对光的红、绿、蓝分量中每个分量进行,将用黑体字来书写包含 RGB 分量的数学形式。

对于(6.6)式和(6.14)式中给出的漫反射和镜面反射公式,通过下面定义的 BRDF,可以进行重写:

$$\rho(\mathbf{V}, \mathbf{L}) = \mathbf{D} + \mathbf{S} \frac{(\mathbf{N} \cdot \mathbf{H})^m}{\mathbf{N} \cdot \mathbf{L}} \quad (6.46)$$

术语“双向”意味着当互换方向向量 \mathbf{V} 和 \mathbf{L} 时,函数 ρ 应该是保持不变,也就是说, ρ 应该满足交换律

$$\rho(\mathbf{V}, \mathbf{L}) = \rho(\mathbf{L}, \mathbf{V}) \quad (6.47)$$

这是为了满足这样的实际情况:在某一传播路径上,将入射光和反射光的方向相互交换后,公式(6.47)不应该产生不同的结果。然而,公式(6.46)给出的函数 ρ 不满足双向要求,因此它在物理上是不正确的。

公式(6.46)违反的另一个物理法则是能量守恒。任何物理上正确的 BRDF 不会在表面的某一点上反射出比入射光更多的光。可以把 BRDF ρ 给出的反射能量分解成两个分量:漫反射能和镜面反射能。分解公式为

$$\rho(\mathbf{V}, \mathbf{L}) = k\mathbf{D} + (1-k)\rho_s(\mathbf{V}, \mathbf{L}) \quad (6.48)$$

其中 \mathbf{D} 是表面的漫反射颜色, k 表示入射光被漫反射的比例,而入射光剩下的 $1-k$ 部分或者被吸收或者形成了镜面反射。将在 6.9.2 节讲述这些效应的建模函数 ρ_s 。

6.9.2 Cook - Torrance 光照模型

在 Cook - Torrance 模型中,将表面看作由称为“微平面”的微小平面组成,这样可以产生真实的镜面反射。每一个微平面被看作一个理想的反射镜,它遵循电磁学理论的反射法则。表面的粗糙度由微平面的斜率来衡量。如图 6.20 所示,一个粗糙表面由一系列斜率变化很大的微平面组成,而组成相对平滑的表面的微平面的斜率则很小。



图 6.20 微平面的斜率变化程度标志了表面的粗糙度

Cook 和 Torrance 用下面的公式计算由公式(6.48)给出的 BRDF 的镜面反射分量 ρ_s ,为

$$\rho_s(\mathbf{V}, \mathbf{L}) = F(\mathbf{V}, \mathbf{L}) \frac{D(\mathbf{V}, \mathbf{L})G(\mathbf{V}, \mathbf{L})}{(\mathbf{N} \cdot \mathbf{V})(\mathbf{N} \cdot \mathbf{L})} \quad (6.49)$$

F 是 Fresnel 系数,它是关于入射角的函数,给出了反射光的量; D 是微平面分布函数,返回的是在给定方向上的微平面的分数值; G 是几何衰减系数,对应于微平面的自身遮蔽阴影。由于微平面是理想的反射面,因此只有那些法向量与平分向量 H 方向相同的微平面才会对镜面反射有所贡献。

6.9.3 Fresnel 系数

电磁波和表面的相互作用,形成了反射波和传导波。反射波所包含的能量等于入射波所包含的能量减去传导波(它被不透明的材质迅速地吸收)所包含的能量。相对于包含法向量 \mathbf{N} 和指向光的方向向量 \mathbf{L} 的平面,入射光的电场可以分解为极化分量,平行于这个平面的分量称为 p 极化分量,垂直于这个平面的分量称为 s 极化分量。对于某一单一波长, Fresnel 系数给出的反射光的 F_p 和 F_s 分数值为

$$\begin{aligned} F_p &= \frac{\tan^2(\theta_1 - \theta_2)}{\tan^2(\theta_1 + \theta_2)} \\ F_s &= \frac{\sin^2(\theta_1 - \theta_2)}{\sin^2(\theta_1 + \theta_2)} \end{aligned} \quad (6.50)$$

其中 θ_1 是入射角, θ_2 是与波长相关的传送角。对于非极化的光,只要求出这两个公式的平均值,就可以得到对应于波长 λ 的 Fresnel 系数 F_λ 。

$$F_\lambda = \frac{1}{2} \left[\frac{\tan^2(\theta_1 - \theta_2)}{\tan^2(\theta_1 + \theta_2)} + \frac{\sin^2(\theta_1 - \theta_2)}{\sin^2(\theta_1 + \theta_2)} \right] \quad (6.51)$$

由于形成镜面反射的每个微平面的朝向使得它的法向量与平分向量 \mathbf{H} 保持方向一致,因此入射角 θ_1 等于 $\cos^{-1}(\mathbf{L} \cdot \mathbf{H})$ 。应用三角公式和 Snell 法则,可以根据 $\mathbf{L} \cdot \mathbf{H}$ 和两种材质的折射度指数 η_1 和 η_2 给出计算 Fresnel 系数的公式。从公式(6.51)中提取出正弦函数可以得到

$$F_{\lambda} = \frac{1}{2} \frac{\sin^2(\theta_1 - \theta_2)}{\sin^2(\theta_1 + \theta_2)} \left[\frac{\cos^2(\theta_1 - \theta_2)}{\cos^2(\theta_1 + \theta_2)} + 1 \right] \quad (6.52)$$

使用角的和、差三角公式将正弦系数展开可以得到

$$\frac{\sin(\theta_1 - \theta_2)}{\sin(\theta_1 + \theta_2)} = \frac{\sin\theta_1\cos\theta_2 - \cos\theta_1\sin\theta_2}{\sin\theta_1\cos\theta_2 + \cos\theta_1\sin\theta_2} = \frac{\eta_{\lambda}\cos\theta_2 - \cos\theta_1}{\eta_{\lambda}\cos\theta_2 + \cos\theta_1} \quad (6.53)$$

其中根据 Snell 法则求得 η_{λ}

$$\eta_{\lambda} = \frac{\eta_2}{\eta_1} = \frac{\sin\theta_1}{\sin\theta_2} \quad (6.54)$$

用 $\cos\theta_1$ 和 η 来表示 $\cos\theta_2$, 首先要把 Snell 法则写成

$$\eta_1 \sqrt{1 - \cos^2\theta_1} = \eta_2 \sqrt{1 - \cos^2\theta_2} \quad (6.55)$$

然后解出 $\cos\theta_2$

$$\cos\theta_2 = \sqrt{1 - \frac{1}{\eta_{\lambda}^2}(1 - \cos^2\theta_1)} \quad (6.56)$$

将变量 g 定义为

$$g = \eta_{\lambda}\cos\theta_2 = \sqrt{\eta_{\lambda}^2 - 1 + (\mathbf{L} \cdot \mathbf{N})^2} \quad (6.57)$$

这样可以将正弦函数的商表示如下:

$$\frac{\sin(\theta_1 - \theta_2)}{\sin(\theta_1 + \theta_2)} = \frac{g - \mathbf{L} \cdot \mathbf{H}}{g + \mathbf{L} \cdot \mathbf{H}} \quad (6.58)$$

按照类似的步骤,可以用 g 和 $\mathbf{L} \cdot \mathbf{H}$ 表示余弦系数。用角的和、差三角公式将余弦展开可得

$$\frac{\cos(\theta_1 + \theta_2)}{\cos(\theta_1 - \theta_2)} = \frac{\cos\theta_1\cos\theta_2 - \sin\theta_1\sin\theta_2}{\cos\theta_1\cos\theta_2 + \sin\theta_1\sin\theta_2} = \frac{\cos\theta_1\cos\theta_2 - \eta_{\lambda}\sin^2\theta_2}{\cos\theta_1\cos\theta_2 + \eta_{\lambda}\sin^2\theta_2} \quad (6.59)$$

代入公式(6.57)中定义的变量 g

$$\begin{aligned} \frac{\cos(\theta_1 + \theta_2)}{\cos(\theta_1 - \theta_2)} &= \frac{g\cos\theta_1 - \eta_{\lambda}^2(1 - \cos^2\theta_2)}{g\cos\theta_1 + \eta_{\lambda}^2(1 - \cos^2\theta_2)} = \frac{g\cos\theta_1 - \eta_{\lambda}^2 + g^2}{g\cos\theta_1 + \eta_{\lambda}^2 + g^2} \\ &= \frac{(\mathbf{L} \cdot \mathbf{H})(g + \mathbf{L} \cdot \mathbf{H}) - 1}{(\mathbf{L} \cdot \mathbf{H})(g - \mathbf{L} \cdot \mathbf{H}) + 1} \end{aligned} \quad (6.60)$$

这样, Fresnel 系数能够完全用 $\mathbf{L} \cdot \mathbf{H}$ 和 η_{λ} 来表示, 公式如下:

$$F_{\lambda}(\mathbf{V}, \mathbf{L}) = \frac{1}{2} \frac{(g - \mathbf{L} \cdot \mathbf{H})^2 \left(\left[\frac{(\mathbf{L} \cdot \mathbf{H})(g + \mathbf{L} \cdot \mathbf{H}) - 1}{(\mathbf{L} \cdot \mathbf{H})(g - \mathbf{L} \cdot \mathbf{H}) + 1} \right]^2 + 1 \right)}{(g + \mathbf{L} \cdot \mathbf{H})^2} \quad (6.61)$$

RGB 颜色的 Fresnel 系数 $F(\mathbf{V}, \mathbf{L})$ 由红、绿、蓝波长的函数 $F_{\lambda}(\mathbf{V}, \mathbf{L})$ 构成。

对于函数 F_{λ} , 可以得出下面两个结论: 第一, 随着入射角趋近于 90° , $\mathbf{L} \cdot \mathbf{H}$ 的值趋近于零, 因此 F_{λ} 的值趋近于 1, 这意味着在擦地入射(入射角为 90°)时, 所有的人射光都被反射, 没有被表面吸收的部分; 第二, 对于从法线入射即入射角为零的情况, $\mathbf{L} \cdot \mathbf{H}$ 的值等于 1, 而且 F_{λ} 的值为

$$(F_{\lambda})_{\mathbf{L}=\mathbf{H}} = \left(\frac{\eta_{\lambda} - 1}{\eta_{\lambda} + 1} \right)^2 \quad (6.62)$$

如果只知道材质的顺着法线入射时的镜面反射颜色 S , 那么上式就给出了一个求 η_{λ} 的近似值的简便方法。从公式(6.62)求解 η_{λ} 可得

$$\eta_{\lambda} = \frac{1 + \sqrt{(F_{\lambda})_{L=H}}}{1 - \sqrt{(F_{\lambda})_{L=H}}} \quad (6.63)$$

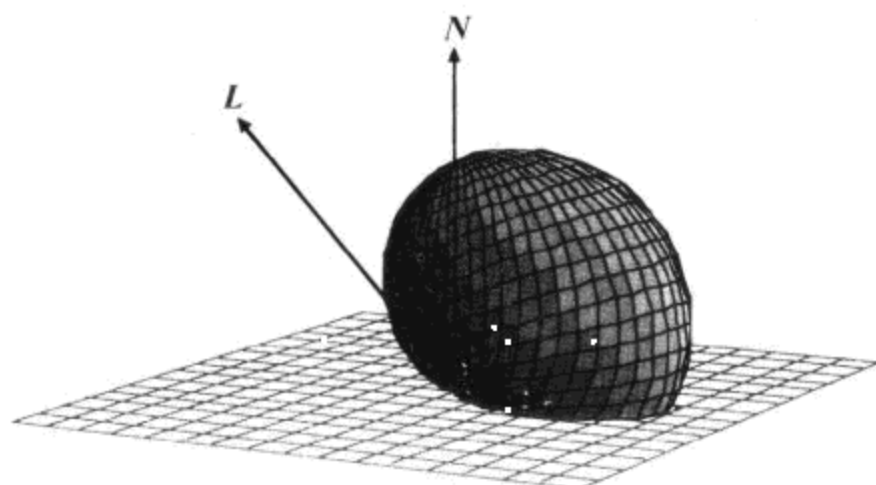
令对应于红、绿、蓝波长的 $(F_{\lambda})_{L=H}$ 值等于 S 的红、绿、蓝分量的值,根据上式就可以计算出 η_{λ} 的值。一旦计算得到 η_{λ} 的值,就可以将它用于公式(6.61),从而计算得到以其他角度入射的光的反射。

6.9.4 微平面分布函数

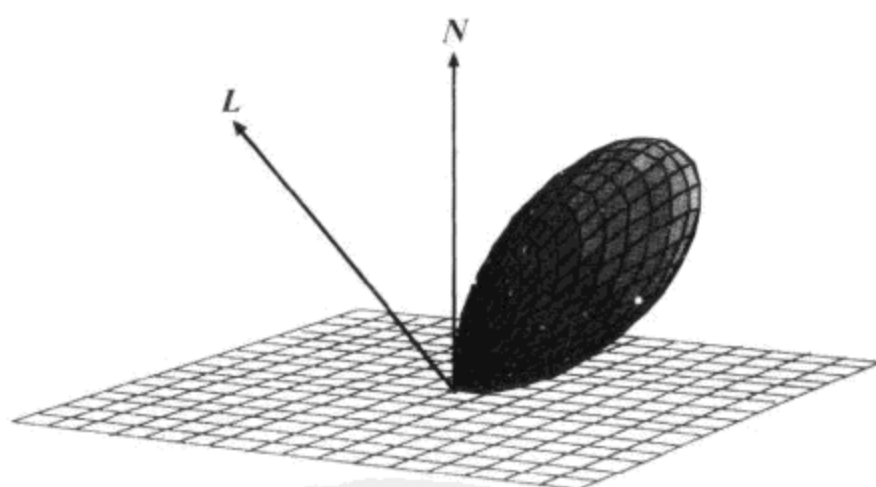
对于给定的一个平分向量 H ,微平面分布函数返回微平面的分数值,其中微平面的法向量的方向与 H 保持一致。对于粗糙表面,Backmann分布函数为

$$D_m(\mathbf{V}, \mathbf{L}) = \frac{1}{4\pi m^2 (\mathbf{N} \cdot \mathbf{H})^4} e^{\frac{(\mathbf{N} \cdot \mathbf{H})^2 - 1}{m^2 (\mathbf{N} \cdot \mathbf{H})^2}} \quad (6.64)$$

该公式给出了用均方根斜率 m 表示的微平面的指向的分布情况。较大的 m 值对应于粗糙平面,此时微平面指向的分布范围很广。如图6.21所示,较小的 m 值对应于较光滑的表面,微平面指向的分布范围相对较小,此时产生相对明显的镜面反射。



(a) 粗糙表面 ($m=0.6$)



(b) 相对光滑的表面 ($m=0.25$)

图 6.21 由公式(6.64)所给出的微平面分布所对应的不同表面模型

公式(6.64)给出的函数是各向同性的,即绕法向量 N 旋转时其值保持恒定不变。如果指向观察者的向量 V 和指向光源的向量 L 之间的夹角保持不变,并且向量 V 、向量 L 和法向量

之间的角也保持不变,那么微平面的分布也是恒定不变的。但是,在实际中很多表面在不同的方向上有着不同的粗糙度,具有这种属性的表面称为各向异性的反射面,打毛的金属、头发和一些特殊的织物就属于该种情况。

修改公式(6.64),通过修改微平面的分布函数来描述各向异性的表面粗糙度

$$D_m(\mathbf{V}, \mathbf{L}) = \frac{1}{4\pi m_x m_y (\mathbf{N} \cdot \mathbf{H})^4} e^{\left(\frac{(\mathbf{T} \cdot \mathbf{P})^2}{m_x^2} + \frac{1 - (\mathbf{T} \cdot \mathbf{P})^2}{m_y^2} \right) \frac{(\mathbf{N} \cdot \mathbf{H})^2 - 1}{(\mathbf{N} \cdot \mathbf{H})^2}} \quad (6.65)$$

其中 m 是二维粗糙度向量, \mathbf{T} 是位于粗糙度为 m_x 的方向上的表面的切线, \mathbf{P} 为平分向量 \mathbf{H} 在切平面上的规格化投影

$$\mathbf{P} = \frac{\mathbf{H} - (\mathbf{N} \cdot \mathbf{H})\mathbf{N}}{\|\mathbf{H} - (\mathbf{N} \cdot \mathbf{H})\mathbf{N}\|} \quad (6.66)$$

图 6.22 显示了分别用各向同性与各向异性的表面粗糙度进行渲染的圆盘。每幅图的 $m_y = 0.1$, 从左到右 m_x 的值分别为 0.1(对应于各向同性)、0.12、0.15 和 0.2。切向量顺着围绕圆盘中心的同心圆,在表面的每个点上垂直于半径。

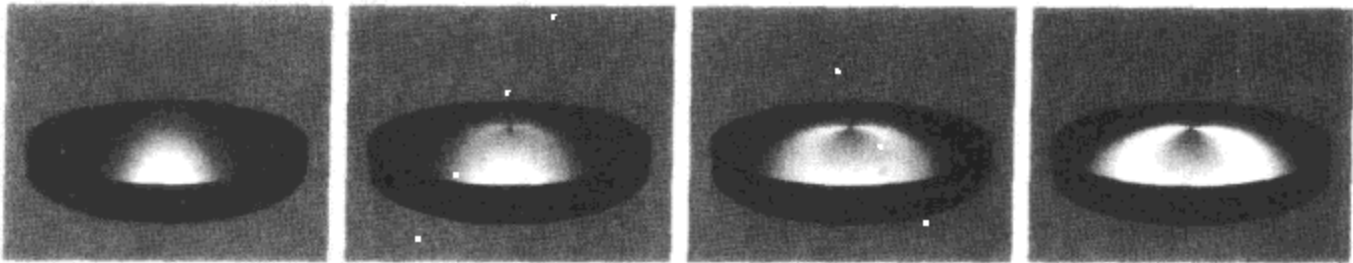


图 6.22 用公式(6.65)给出的各向异性的分布函数进行渲染的圆盘

许多表面会显示出不同尺度的粗糙效果,为了得到这样的效果可以计算多个微平面分布函数的加权平均值为

$$D(\mathbf{V}, \mathbf{L}) = \sum_{i=1}^n w_i D_{m_i}(\mathbf{V}, \mathbf{L}) \quad (6.67)$$

其中用到了不同尺度的粗糙度值 m_i , 所有权值 w_i 的和为 1。图 6.23 给出了使用不同 m 值渲染的两个物体,第三个物体是用这些值加权和渲染的。前面两幅图像各采用了单一粗糙度值,分别为 $m_1 = 0.1$, $m_2 = 0.25$ 。第三幅图像是前两个粗糙度值的加权和,权值分别为 $w_1 = 0.4$, $w_2 = 0.6$ 。

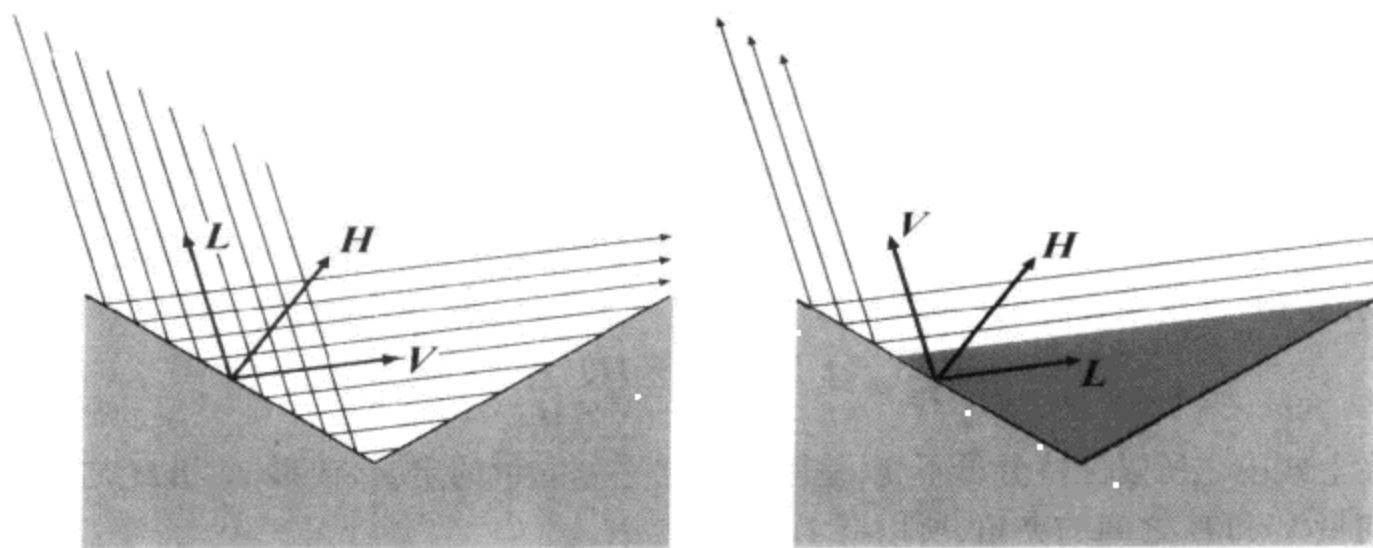


图 6.23 使用各向同性微平面分布渲染得到的铜花瓶

6.9.5 几何衰减系数

微平面上的一些入射光,在到达一个表面之前或被该表面反射之后,可能会被相邻的微平面阻挡。这种阻挡会造成镜面反射的轻微昏暗,可以用几何衰减系数来说明这种效果。被遮挡的光主要在任意方向上发散,最终形成了表面漫反射的一部分。

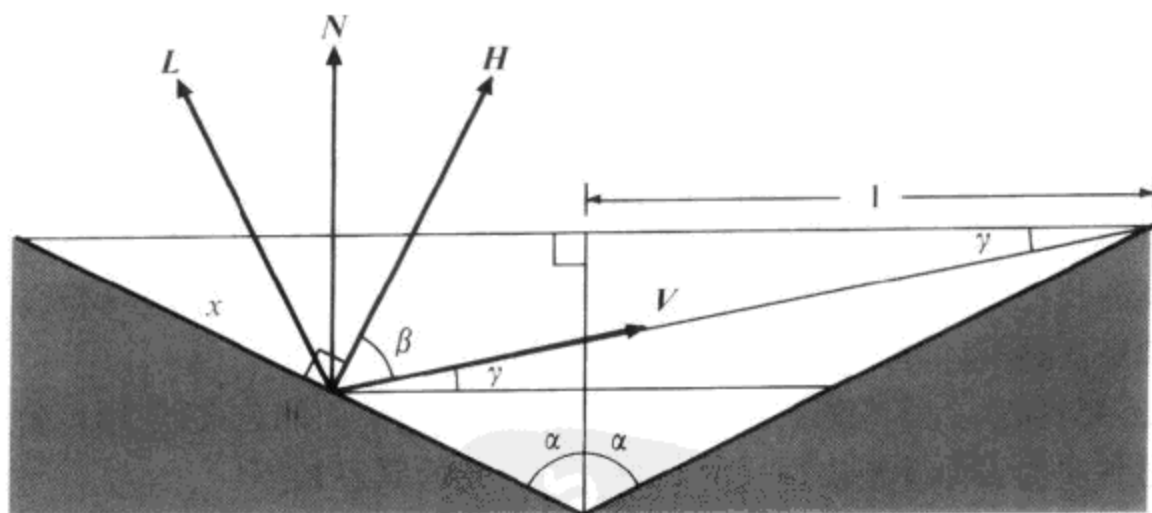
假设微平面之间形成了 V 型槽,这样就可以估算出由于表面的粗糙度会有多少光线被遮挡住。图 6.24(a)说明了从一个微平面反射出来的光有一部分被相邻的微平面遮挡的情况,在这种情况下,光是在被反射以后才被遮挡的。翻转光的传播方向,就会出现光在到达微平面之前被遮挡的情况,如图 6.24(b)所示。



(a) 从左微平面反射出的一部分光被右微平面遮挡 (b) 光在到达左微平面前被右微平面遮挡

图 6.24 光被微平面遮挡的情况

根据一些基本的三角学知识推导出一个公式,按照这个公式就可以计算出被一个微平面反射并被相邻微平面遮挡后仍到达观察者眼中光的分数值。如图 6.25 所示,要确定的是在宽度为 w 的微平面上,观察者可以看到的那部分的值。



注:要求的分数值为 x/w 。平分向量垂直于微平面的表面,因为只有具有该指向的微平面才会影响镜面反射

图 6.25 从左微平面反射出并到达观察者的光的分数值

首先从图 6.25 可以观察到

$$w = \frac{1}{\sin\alpha} \quad (6.68)$$

然后根据正弦定理有

$$x = \frac{2\sin\gamma}{\sin(\beta + \pi/2)} \quad (6.69)$$

可以将公式(6.68)和(6.69)中的正弦函数用平移 $\pi/2$ 弧度的余弦函数来表示

$$\begin{aligned} \sin\alpha &= \cos(\pi/2 - \alpha) = \mathbf{N} \cdot \mathbf{H} \\ \sin(\beta + \pi/2) &= \cos(\beta) = \mathbf{V} \cdot \mathbf{H} \\ \sin\gamma &= \cos(\pi/2 - \gamma) = \mathbf{N} \cdot \mathbf{V} \end{aligned} \quad (6.70)$$

使用每个余弦函数所对应的点积,可以将到达观察者的光的分数值 G_1 表示为

$$G_1 = \frac{x}{w} = \frac{2(\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \mathbf{V})}{\mathbf{V} \cdot \mathbf{H}} \quad (6.71)$$

对于没有到达微平面前就被遮挡的光,只要简单地互换图 6.25 中的向量 \mathbf{V} 和 \mathbf{L} ,就可以得到此时到达观察者的分数值 G_2

$$G_2 = \frac{x}{w} = \frac{2(\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \mathbf{L})}{\mathbf{V} \cdot \mathbf{H}} \quad (6.72)$$

在上式中之所以保持分母不变,是因为根据平分向量的定义,向量 \mathbf{L} 和 \mathbf{H} 之间的夹角等于向量 \mathbf{V} 和 \mathbf{H} 之间的夹角,所以有 $\mathbf{L} \cdot \mathbf{H} = \mathbf{V} \cdot \mathbf{H}$ 。

要被微平面反射的光可能出现三种情况:光未被遮挡(这时到达观察者的光的分数值为 1),部分反射光被遮挡,部分入射光被遮挡。通过将几何衰减系数定义为到达观察者的光的最小分数值,就可以综合表示这三种情况

$$\begin{aligned} G(\mathbf{V}, \mathbf{L}) &= \min\{1, G_1, G_2\} \\ &= \min\left\{1, \frac{2(\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \mathbf{V})}{\mathbf{V} \cdot \mathbf{H}}, \frac{2(\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \mathbf{L})}{\mathbf{V} \cdot \mathbf{H}}\right\} \end{aligned} \quad (6.73)$$

6.9.6 实现

在光线跟踪的应用中,无论何时计算光线与表面的相交,都可以直接使用公式(6.49)。但是,在渲染三角网格时,针对每个像素的计算过程太过复杂。对于划分了足够多方格的表面,针对每个顶点计算公式(6.49),会得到较好的效果,但是这种方法计算量过大。有一个避免过多运算的技巧,就是使用纹理映射图,在纹理映射图中存储的是查找表,这些查找表是以在每个顶点上容易计算出来的数值作为索引的。

在公式(6.45)加入环境光和发射光的成分,可以得到下面的公式,使用该公式可以求出 n 个光源照射到曲面表面并被表面反射到观察者的光的颜色 \mathbf{K}

$$\mathbf{K} = \mathbf{E} + \mathbf{DA} + \sum_{i=1}^n \rho(\mathbf{V}, \mathbf{L}_i) C_i(\mathbf{N} \cdot \mathbf{L}_i) \quad (6.74)$$

用公式(6.48)代换 BRDF ρ ,可以得到

$$\mathbf{K} = \mathbf{E} + \mathbf{DA} + k\mathbf{D} \sum_{i=1}^n C_i(\mathbf{N} \cdot \mathbf{L}_i) + (1 - k) \sum_{i=1}^n \rho_s(\mathbf{V}, \mathbf{L}_i) C_i(\mathbf{N} \cdot \mathbf{L}_i) \quad (6.75)$$

使用公式(6.49),就可以得出单一光源所产生的 K 的镜面分量

$$\begin{aligned} K_{\text{specular}} &= C(1-k)\rho_s(\mathbf{V},\mathbf{L})(\mathbf{N}\cdot\mathbf{L}) \\ &= C(1-k)F(\mathbf{V},\mathbf{L})\frac{D_m(\mathbf{V},\mathbf{L})G(\mathbf{V},\mathbf{L})}{\mathbf{N}\cdot\mathbf{V}} \end{aligned} \quad (6.76)$$

其中 C 是光的颜色, k 是被漫反射的光的分数值, Fresnel 系数 $F(\mathbf{V},\mathbf{L})$ 只与数量 $\mathbf{L}\cdot\mathbf{H}$ 有关, 各向同性微平面分布函数 $D_m(\mathbf{V},\mathbf{L})$ 只与数量 $\mathbf{N}\cdot\mathbf{H}$ 有关。给定一个法向入射的镜面反射颜色 S 和一个微平面的均方根斜率 m , 就可以构造一个纹理映射图, 它的 s 坐标和 t 坐标分别对应于 $\mathbf{L}\cdot\mathbf{H}$ 和 $\mathbf{N}\cdot\mathbf{H}$, 它的颜色值表示的是乘积 $F(\mathbf{V},\mathbf{L})D_m(\mathbf{V},\mathbf{L})$ 。此外, 还可以将常数 $1-k$ 和几何衰减系数 $G(\mathbf{V},\mathbf{L})$ 的一部分包括在该图中。

令 F 为 $\mathbf{L}\cdot\mathbf{H}$ 的函数, 令 D_m 为 $\mathbf{N}\cdot\mathbf{H}$ 的函数, 可以得到

$$K_{\text{specular}} = C(1-k)F(\mathbf{L}\cdot\mathbf{H})D_m(\mathbf{N}\cdot\mathbf{H})\frac{G(\mathbf{V}\cdot\mathbf{L})}{\mathbf{N}\cdot\mathbf{V}} \quad (6.77)$$

对于系数 $G(\mathbf{V}\cdot\mathbf{L})/\mathbf{N}\cdot\mathbf{V}$, 将公式(6.73)中的 $\mathbf{V}\cdot\mathbf{H}$ 用 $\mathbf{L}\cdot\mathbf{H}$ 替换, 并代入公式(6.77)中可得

$$\frac{G(\mathbf{L}\cdot\mathbf{H})}{\mathbf{N}\cdot\mathbf{V}} = \min\left\{\frac{1}{\mathbf{N}\cdot\mathbf{V}}, \frac{2(\mathbf{N}\cdot\mathbf{H})}{\mathbf{L}\cdot\mathbf{H}}, \frac{2(\mathbf{N}\cdot\mathbf{H})(\mathbf{N}\cdot\mathbf{L})}{(\mathbf{L}\cdot\mathbf{H})(\mathbf{N}\cdot\mathbf{V})}\right\} \quad (6.78)$$

这里已经假设 $\mathbf{N}\cdot\mathbf{V}$ 为正值, 否则表面将无法看到。由于公式(6.78)的右半部分没有上界, 所以从最小值函数中提取 $2(\mathbf{N}\cdot\mathbf{H})/\mathbf{L}\cdot\mathbf{H}$ 来限定其范围。

$$\frac{G(\mathbf{V},\mathbf{L})}{\mathbf{N}\cdot\mathbf{V}} = \frac{2(\mathbf{N}\cdot\mathbf{H})}{\mathbf{L}\cdot\mathbf{H}} \min\left\{\frac{\mathbf{L}\cdot\mathbf{H}}{2(\mathbf{N}\cdot\mathbf{H})(\mathbf{N}\cdot\mathbf{V})}, 1, \frac{\mathbf{N}\cdot\mathbf{L}}{\mathbf{N}\cdot\mathbf{V}}\right\} \quad (6.79)$$

现在, 整个镜面分量可表示为

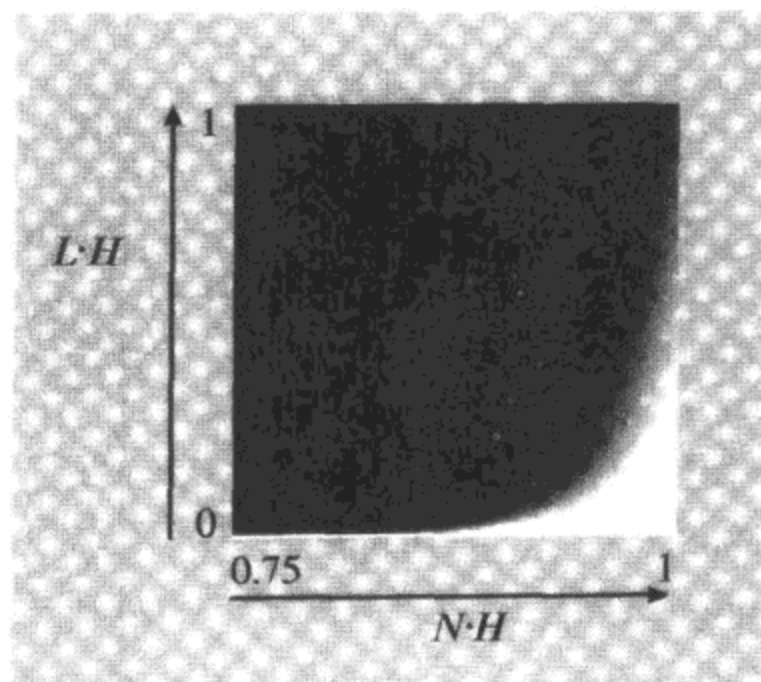
$$K_{\text{specular}} = 4[CX(\mathbf{V},\mathbf{L})][Y(\mathbf{N}\cdot\mathbf{H})][Z(\mathbf{L}\cdot\mathbf{H})] \quad (6.80)$$

其中函数 X , Y 和 Z 分别为

$$\begin{aligned} X(\mathbf{V}\cdot\mathbf{L}) &= \min\left\{\frac{\mathbf{L}\cdot\mathbf{H}}{2(\mathbf{N}\cdot\mathbf{H})(\mathbf{N}\cdot\mathbf{V})}, 1, \frac{\mathbf{N}\cdot\mathbf{L}}{\mathbf{N}\cdot\mathbf{V}}\right\} \\ Y(\mathbf{N}\cdot\mathbf{H}) &= (1-k)(\mathbf{N}\cdot\mathbf{H})D_m(\mathbf{N}\cdot\mathbf{H}) \\ Z(\mathbf{L}\cdot\mathbf{H}) &= \frac{F(\mathbf{L}\cdot\mathbf{H})}{2(\mathbf{L}\cdot\mathbf{H})} \end{aligned} \quad (6.81)$$

若 $\mathbf{N}\cdot\mathbf{L}$, $\mathbf{N}\cdot\mathbf{V}$, $\mathbf{L}\cdot\mathbf{H}$ 中的任何一个值小于零, 就将 X 设为零。否则, X 的值会小于 1, 这时就计算三角网格的每个顶点的 CX , 并将计算得到的值存入颜色数组中。程序 6.2 示例了如何在顶点程序中实现这些计算。乘积 YZ 先于 $\mathbf{L}\cdot\mathbf{H}$ 和 $\mathbf{N}\cdot\mathbf{H}$ 计算, 用于构造类似于图 6.26 所示的纹理图。由于函数 Z 分母中的 2 缩小了乘积的大小, 所以纹理图可以表示更大范围的值。用纹理图的一个取样乘以颜色数组所提供的 CX 的插值, 将结果放大 4 倍, 以生成用于像素明暗处理的最终的镜面反射颜色。图 6.26 中 YZ 的系数由公式(6.81)给出。 s 坐标对应于值 $4(\mathbf{N}\cdot\mathbf{H})-3$, 而 t 坐标对应于值 $\mathbf{L}\cdot\mathbf{H}$ 。这幅图像的生成, 用到了法向入射的镜面反射颜色 $S = \langle 0.8, 0.6, 0.1 \rangle$, 微平面均方根斜率 $m = 0.25$, 以及漫反射分数值 $k = 0.6$ 。

【程序 6.2】 这个顶点程序完成了实现各向同性 Cook-Torrance 光照所需的基本计算。主颜色设定为 CX , 纹理单元 0 的坐标值为 $\langle 4(\mathbf{N}\cdot\mathbf{H})-3, \mathbf{L}\cdot\mathbf{H} \rangle$ 。

图 6.26 表示乘积 YZ 的纹理图

```
!!VP1.0
```

```
# c[0-3] = 投影 * 模型图 矩阵
# c[4] = 常量 (1.0, 4.0, 3.0, 0.0)
# c[5] = 物体空间中摄像机的位置
# c[6] = 物体空间中光的方向 L
# c[7] = 光的颜色 C
```

```
DP4 o[HPOS].x, c[0], v[OPOS]; # 变换顶点
DP4 o[HPOS].y, c[1], v[OPOS];
DP4 o[HPOS].z, c[2], v[OPOS];
DP4 o[HPOS].w, c[3], v[OPOS];

MOV R0, v[NRML]; # 装入法向量 N

ADD R2, c[5], -v[OPOS]; # R2= 视方向 V
DP3 R5, R2, R2; # 规格化 V
RSQ R5.x, R5.x;
MUL R2.xyz, R2, R5.x;

ADD R3, R2, c[6]; # R3= 平分向量 H
DP3 R6, R3, R3; # 规格化 H
RSQ R6.x, R6.x;

MUL R3.xyz, R3, R6.x;
```

```

DP3   R4.x, R0, c[6];           # R4.x = N · L
DP3   R4.y, R0, R2;           # R4.y = N · V
DP3   R4.z, R0, R2;           # R4.z = N · H
DP3   R4.w, c[6], R3;         # R4.w = L · H

ADD   R5.x, R4.z, R4.z;       # R5.x = 2 (N*H)
RCP   R5.x, R5.x;             # R5.x = 1/(2(N*H))
RCP   R5.y, R4.y;           # R5.y = 1/(N*V)
MUL   R6.x, R5.x, R4.w;       # R6.x = (L*H)/(2 (N*H))
MUL   R6.x, R6.x, R5.y;       # R6.x = (L*H)/(2 (N*H)(N*V))
MUL   R6.y, R5.y, R4.x;       # R6.y = (N*L)/(N*V)

MIN   R5.x, R6.x, R6.y;       # 计算最小值
MIN   R5.x, R5.x, c[4].x;     # R5.x= min(R5.x, 1)

SLT   R6, -R4, c[4].w;       # 点积是否大于0?
MUL   R6.x, R6.x, R6.y;       # R6.x = (N*L > 0) (N*V > 0)
MUL   R6.x, R6.x, R6.w;       # R6.x *= (L*H > 0)
MUL   R5.x, R5.x, R6.x;       # R5.x *= 0 1

# 各向异性时, 下面的代码要更改
MUL   o[COLO], R5.x, c[7];     # R5.x * 光的颜色
MAD   o[TEX0].x, R4.z, c[4].y, -c[4].z; # s = 4(N*H)-3
MOV   o[TEX0].y, R4.w;         # t = L*H
END

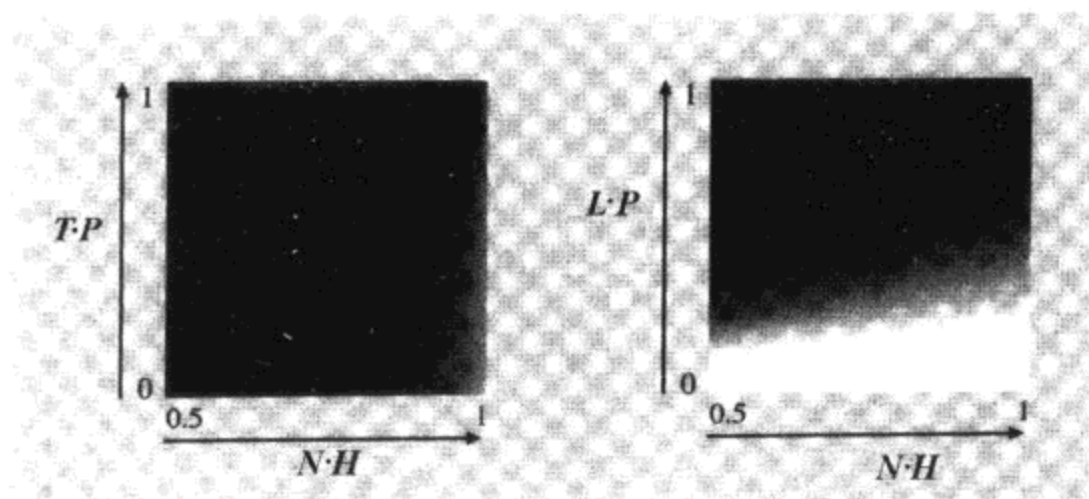
```

对于各向异性的表面,微平面分布函数 D_m 取决于点积 $T \cdot P$ 和 $N \cdot H$ 。在这种情况下,将 Y 和 Z 定义如下:

$$Y(N \cdot H, T \cdot P) = \frac{1-k}{\sqrt{2}} D_m(N \cdot H, T \cdot P)$$

$$Z(N \cdot H, L \cdot H) = \frac{(N \cdot H)F(L \cdot H)}{\sqrt{2}(L \cdot H)} \quad (6.82)$$

原来出现在 Z 的分母中的 2,现在开根号后分别出现在 Y 和 Z 中。三维纹理图能用做 YZ 积的查找表,但是对于不支持三维纹理图的硬件,则需要将 Y 和 Z 分别存储到与图 6.27 所示的纹理图相似的二维纹理图中。程序 6.3 给出了对各向同性的顶点程序所作的修改,以适应各向异性反射的情况。图 6.27 中的 s 坐标对应于值 $2(N \cdot H) - 1$ 。左图中的 t 坐标对应于值 $T \cdot P$,右图中的 t 坐标对应于值 $L \cdot H$ 。除了均方根斜率被改成各向异性 ($m = \langle 0.25, 0.5 \rangle$) 外,产生这幅图所用到的其他值均与图 6.26 的相同。

图 6.27 由公式(6.82)所给出的 Y 值(左图)和 Z 值(右图)所对应的纹理图

【程序 6.3】 用下面代码代替程序 6.2 的最后四行,可完成各向异性的 Cook-Torrance 光照的必要计算。顶点的切线向量从属性数组 6 中读出。纹理映射坐标 $\langle 2(N \cdot H) - 1, L \cdot H \rangle$ 存储到纹理单元 0,坐标存储 $\langle 2(N \cdot H) - 1, T \cdot P \rangle$ 到纹理单元 1。

```
# c[4] = constant (1.0, 2.0, 0.0, 0.0)

MAD R7, -R4.z, R0, R3;          # R7 = P = H - (N*H)N
DP3 R8, R7, R7;                # 正规化 P
RSQ R8.x, R8.x;
MUL R7.xyz, R7, R8.x;

MUL o[COLO], R5.x, C[7];        # R5.x * 光的颜色
MAD o[TEX0].x, R4.z, c[4].y, -c[4].x; # s0 = 2(N*H)-1
MOV o[TEX0].y, R4.w;           # t0 = L*H
MAD o[TEX1].x, R4.z, c[4].y, -c[4].x; # s1 = 2(N*H)-1
DP3 o[TEX1].y, v[6], R7;       # t1 = T*P
END
```

6.10 本章小结

点光源的衰减

与光源位置的距离为 d 处的光强为

$$C = \frac{1}{k_c + k_l d + k_q d^2} C_0$$

其中的 C_0 为光的颜色,衰减由常数 k_c, k_l 和 k_q 控制。

聚焦光源的衰减

与光源位置的距离为 d 的点 Q 处的聚焦光源的光强 C 为

$$C = \frac{\max\{-U \cdot L, 0\}^p}{k_c + k_l d + k_q d^2} C_0$$

其中, C_0 为光的颜色, k_c , k_l 和 k_q 是衰减常数, U 为聚焦光所指的方向, L 是从 Q 指向光源的单位向量; 指数 p 控制光强随 U 和 $-L$ 之间的角度的增加而下降的速率。

环境光照和漫反射光照

在表面的 Q 点, 环境光和漫反射光对光照颜色的影响如下式:

$$K_{\text{diffuse}} = DA + D \sum_{i=1}^n C_i \max\{N \cdot L_i, 0\}$$

其中, D 是表面的漫反射颜色, N 是表面的法向量, L_i 是从 Q 点指向第 i 个光源的单位向量, C_i 是 Q 点处第 i 个光源的光强, A 为环境光的颜色。

镜面反射光照

在表面 Q 点, 镜面反射光对影响光照颜色的影响如下式:

$$K_{\text{specular}} = S \sum_{i=1}^n C_i \max\{N \cdot H_i, 0\}^m (N \cdot L_i > 0)$$

其中, S 是表面的镜面反射颜色, H_i 是在 Q 点处位于指向光 L_i 的向量和指向观察者的向量之间的单位平分向量; m 控制镜面反射的聚集程度。表达式 $(N \cdot L_i > 0)$ 的值为 1 或 0, 取决于表面是否朝向入射光。

整体光照公式

由 n 束光照射的表面点 Q 处的反射光颜色 K 的计算式为

$$K = EM + DAT + \sum_{i=1}^n C_i [DT(N \cdot L_i) + SG(N \cdot H_i)^m (N \cdot L_i > 0)]$$

其中, 点积 $N \cdot L$ 和 $N \cdot H$ 固定为 0, 其他涉及的量定义如下:

D = 漫反射颜色

S = 镜面反射颜色

m = 镜面反射指数

A = 环境光的颜色

E = 发射光的颜色

T = 纹理图的颜色

G = 光泽图的颜色

M = 发射图的颜色

C_i = Q 点处第 i 束光的颜色

L_i = 第 i 束光的指向光的方向向量

H_i = 第 i 束光的平分向量

$N =$ 法向量

凹凸映射

以 E, F, G 为顶点的三角形的切线 T 和副法线 B 的计算公式为

$$\begin{bmatrix} T_x & T_y & T_z \\ B_x & B_y & B_z \end{bmatrix} = \frac{1}{s_1 t_2 - s_2 t_1} \begin{bmatrix} t_2 & -t_1 \\ -s_2 & s_1 \end{bmatrix} \begin{bmatrix} P_x & P_y & P_z \\ Q_x & Q_y & Q_z \end{bmatrix}$$

其中 $P = F - E, Q = G - E$, 并且有

$$\langle s_1, t_1 \rangle = \langle s_F - s_E, t_F - t_E \rangle$$

$$\langle s_2, t_2 \rangle = \langle s_G - s_E, t_G - t_E \rangle$$

指向光的方向向量 L 和平分向量 H 用下面的矩阵从对象空间转换到切线空间:

$$\begin{bmatrix} T'_x & T'_y & T'_z \\ B'_x & B'_y & B'_z \\ N_x & N_y & N_z \end{bmatrix}$$

其中 T' 和 B' 正交于 N , 并且相互正交。

双向反射分布函数

被 n 束光照射的表面沿着方向 V 的反射光的辐射强度 C_R 的计算公式为

$$C_R(V) = \sum_{i=1}^n \rho(V, L_i) C_i(N \cdot L_i)$$

其中 C_i 为第 i 个光源的辐射强度。BRDF ρ 可以分为漫反射分量和镜面反射分量, 表示如下:

$$\rho(V, L) = kD + (1 + k)\rho_s(V, L)$$

其中, k 为漫反射光的分数值。

Cook - Torrance 光照

用在 Cook - Torrance 光照模型里的 BRDF 的镜面反射分量的计算公式为

$$\rho_s(V, L) = F(V, L) \frac{D(V, L)G(V, L)}{(N \cdot V)(N \cdot L)}$$

其中 F 是 Fresnel 系数, D 是微平面分布函数, G 是几何衰减系数。

Fresnel 系数

单颜色的 Fresnel 系数的计算公式为

$$F_\lambda(V, L) = \frac{1}{2} \frac{(g - L \cdot H)^2}{(g + L \cdot H)^2} \left[\frac{[(L \cdot H)(g + L \cdot H) - 1]^2}{[(L \cdot H)(g - L \cdot H) + 1]^2} + 1 \right]$$

其中, g 的定义如下:

$$g = \sqrt{\eta_\lambda^2 - 1 + (L \cdot N)^2}$$

折射率 η_λ 可用以下公式计算:

$$\eta_{\lambda} = \frac{1 + \sqrt{S_{\lambda}}}{1 - \sqrt{S_{\lambda}}}$$

其中, S 是法线入射情况下的镜面反射颜色。

微平面分布函数

各向同性的微平面分布函数为

$$D_m(\mathbf{V}, \mathbf{L}) = \frac{1}{4\pi m^2 (\mathbf{N} \cdot \mathbf{H})^4} e^{\frac{(\mathbf{N} \cdot \mathbf{H})^2 - 1}{m^2 (\mathbf{N} \cdot \mathbf{H})^2}}$$

其中 m 是微平面的均方根斜率。对于各向异性的表面, 微平面分布函数为

$$D_m(\mathbf{V}, \mathbf{L}) = \frac{1}{4\pi m_x m_y (\mathbf{N} \cdot \mathbf{H})^4} e^{\left(\frac{(\mathbf{T} \cdot \mathbf{P})^2}{m_x^2} + \frac{1 - (\mathbf{T} \cdot \mathbf{P})^2}{m_y^2} \right) \frac{(\mathbf{N} \cdot \mathbf{H})^2 - 1}{(\mathbf{N} \cdot \mathbf{H})^2}}$$

其中 m_x 和 m_y 表示相互平行且垂直于切线方向 \mathbf{T} 的均方根斜率。向量 \mathbf{P} 是平分向量 \mathbf{H} 在切平面上的规格化投影。

几何衰减系数

几何衰减系数的表达式为

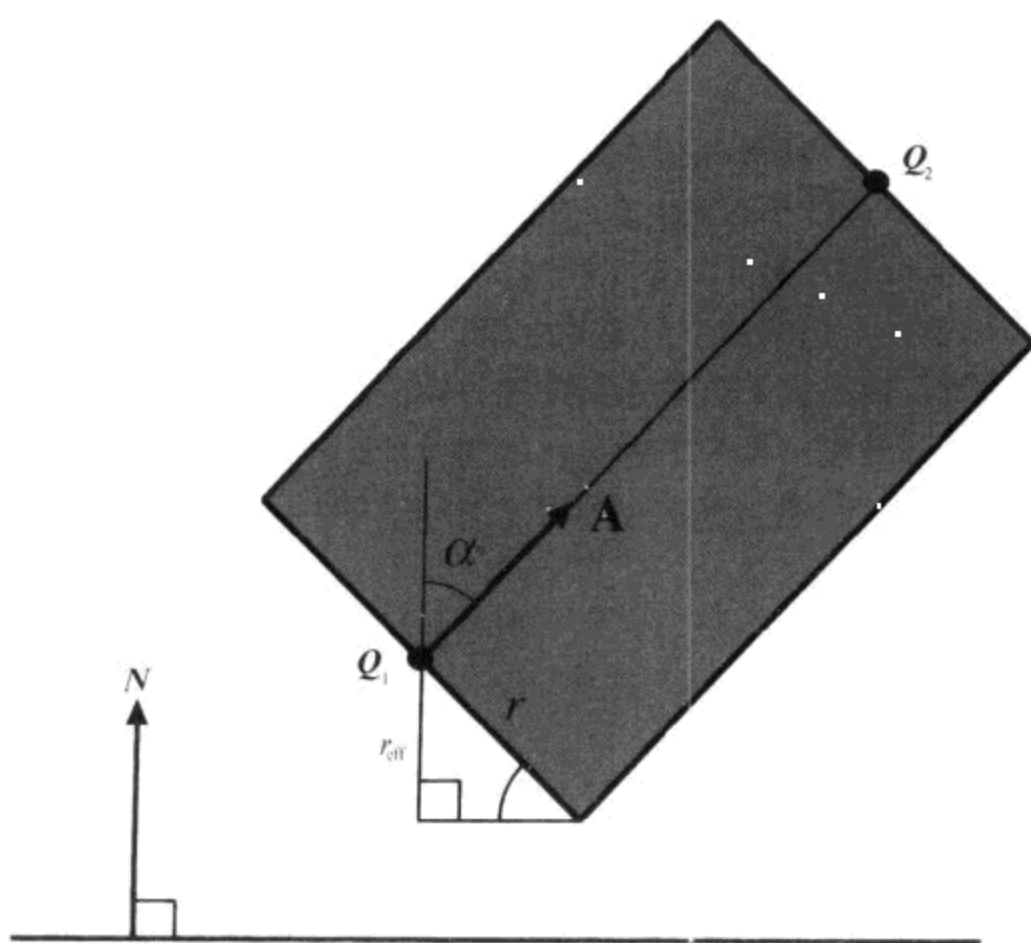
$$G(\mathbf{V}, \mathbf{L}) = \min \left\{ 1, \frac{2(\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \mathbf{V})}{\mathbf{V} \cdot \mathbf{H}}, \frac{2(\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \mathbf{L})}{\mathbf{V} \cdot \mathbf{H}} \right\}$$

该公式解释了微平面的人射光或反射光受到相邻微平面遮挡的现象。

习 题

1. 一点光源的衰减常数为 $k_c = 1, k_l = 0, k_q = \frac{1}{2}$ 。问距光源多远处的光强度是距光源 1 米处的光强度的 1/4?
2. 有一聚焦光源位于原点上方 10 米处的点 $\mathbf{P} = \langle 0, 0, 10 \rangle$, 在方向 $\mathbf{U} = \langle 0, 0, -1 \rangle$ 上的辐射能量因为设置了 $k_c = 1, k_l = k_q = 0$ 而不会发生距离衰减。如果光的颜色为白色 ($\mathbf{C}_0 = \langle 1, 1, 1 \rangle$), 聚焦指数为 8, 那么在 $x-y$ 平面上的圆的半径为多大时, 该圆上的光强为 50% 的灰度 ($\mathbf{C} = \langle 1/2, 1/2, 1/2 \rangle$)?
3. 描述当 $\mathbf{N} \cdot \mathbf{L}$ 为负数时, $\mathbf{N} \cdot \mathbf{H}$ 怎样才能为正数, 以此判断光照公式中的条件 ($\mathbf{N} \cdot \mathbf{L} > 0$) 的必要性。
4. 编写一个程序, 要求该程序能计算任意三角网格的顶点法线和顶点切线。假设已经对三角网格做了这样的设定: 对于 n 个三角形中的任意一个, 都可以在存储了 m 个顶点的数组中索引到三个顶点的信息, 顶点数组中存储的每一顶点信息包含了该顶点的位置坐标和二维纹理映射坐标。
5. 实现一个简单的光线跟踪器, 能够用公式 (6.6) 和 (6.14) 计算漫反射和镜面反射。要求光线跟踪器能够对球体进行建模, 并且支持定向光源、点光源和聚焦光源。
6. 对第 5 题中的光线跟踪器进行扩展, 实现 Cook-Torrance 光照。

第 7 章 可视性判断



在模型渲染结构中,可视性判断是衡量一个实时三维引擎性能的最重要的依据。任何一个三维引擎必须能够根据给定的相机位置与方向,快速地计算出一个场景中哪些内容是可见的,并对场景中可见部分进行渲染处理。为了解决可视性判断问题,三维引擎通常从该问题的反面着手,即首先判断场景中哪些内容是不可见的,然后对剩下的内容进行渲染处理。大多数三维引擎中的可视性判断分层完成,其主要目的是确定哪一个几何体与视平截面不相交。最小规模的可视性判断是指三维图形硬件执行隐面消除算法剔除某个几何体表面中那些方向远离相机的三角形。稍大规模的可视性判断则是用边界体测试法判断场景中的一个几何体是否完全位于视平截面之外。更大规模的可视性判断是通过将场景中的区域表示成一些树结构,如二叉空间分割树(Binary Space Partitioning, BSP)或八叉树,这样一组几何体可以从可视集合中分离出来。最大规模的可视性判断是指整个几何场景通过所谓的门系统技术被从可视集合中去除。

7.1 边界体构造

构造包含一个三角网格所有顶点的边界体,可以保证网格中的每个三角形都属于一个边界体。边界体应该尽可能小以致它尽可能完全位于视平截面之外,从而确保尽可能地将它包含的几何对象从可视几何体集合中剔除。

图 7.1 所示的是一个包围一个点集盒子,点集中的点是一个三角网格的顶点。该盒子的边与坐标轴平行,而点集中点的分布方式导致该盒子中含有大量的空区间。

而如图 7.2 所示,通过选择一个边与点集的自然轴平行的盒子可以大大减小盒子的尺寸。7.1.1 节将提供一个计算点集自然轴的方法。

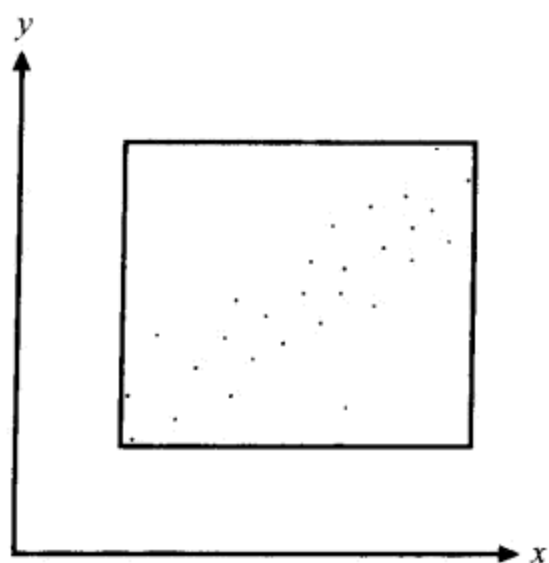


图 7.1 边与坐标轴平行的边界体不适用于大多数的顶点分布

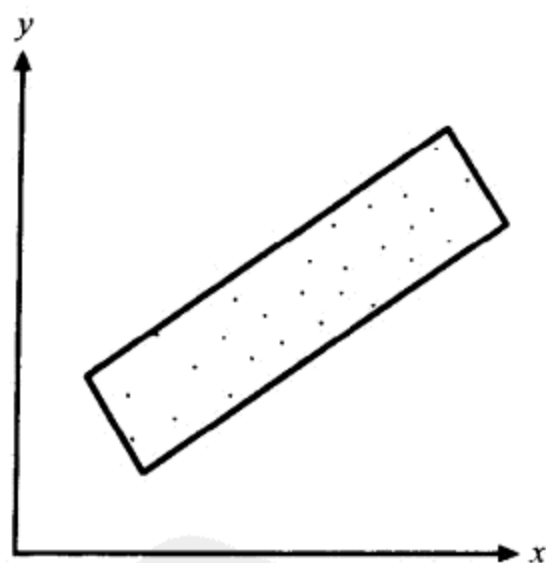


图 7.2 选择合适的平行轴可以构造更好的边界体

7.1.1 主分量分析

通过选择一个坐标轴与三角网格的顶点集自然对齐的坐标系,可以进一步减小相应的边界体尺寸。统计学中的主分量分析方法可用于计算该坐标系,主分量分析方法常用于计算由多个变量组成并且可以被分割成不相关分量的数据集合的坐标空间,保存在一个顶点

位置数组中的 x 、 y 和 z 坐标就属于这种数据集合。数据集合的基本主分量是该集合中数据变化最大的方向。

对于任意的由 N 个顶点 P_1, P_2, \dots, P_N 组成的集合, 其中 $P_i = \langle x_i, y_i, z_i \rangle$, 为了计算它的自然坐标系, 首先要使用以下方程式计算幂(平均)位置的坐标:

$$m = \frac{1}{N} \sum_{i=1}^N P_i \quad (7.1)$$

然后, 构造一个 3×3 的协方差矩阵为

$$C = \frac{1}{N} \sum_{i=1}^N (P_i - m)(P_i - m)^T \quad (7.2)$$

该协方差矩阵是由以下 6 个惟一元素组成的对称矩阵

$$\begin{aligned} C_{11} &= \frac{1}{N} \sum_{i=1}^N (x_i - m_x)^2 & C_{12} &= C_{21} = \frac{1}{N} \sum_{i=1}^N (x_i - m_x)(y_i - m_y) \\ C_{22} &= \frac{1}{N} \sum_{i=1}^N (y_i - m_y)^2 & C_{13} &= C_{31} = \frac{1}{N} \sum_{i=1}^N (x_i - m_x)(z_i - m_z) \\ C_{33} &= \frac{1}{N} \sum_{i=1}^N (z_i - m_z)^2 & C_{23} &= C_{32} = \frac{1}{N} \sum_{i=1}^N (y_i - m_y)(z_i - m_z) \end{aligned} \quad (7.3)$$

这些协方差矩阵中的元素表示每对 x 、 y 和 z 坐标之间的关系。值为 0 的元素表示相应的两个坐标之间没有关系。如果矩阵 C 为对角矩阵, 则对应点的所有 x 、 y 和 z 坐标之间没有关系, 也就是说这些点沿每个坐标轴都是分离的。

为了获得对角协方差矩阵, 需要一个可以对顶点集合进行相应变换的基。如果用变换矩阵 A 对顶点集合 $\{P_i\}$ 中的每个点进行变换, 那么变换后顶点集合的协方差矩阵 C' 为

$$\begin{aligned} C' &= \frac{1}{N} \sum_{i=1}^N (AP_i - Am)(AP_i - Am)^T \\ &= \frac{1}{N} \sum_{i=1}^N A(P_i - m)(P_i - m)^T A^T \\ &= ACA^T \end{aligned} \quad (7.4)$$

因此, 需要一个转置矩阵可以将矩阵 C 对角化的正交矩阵 A 。由于矩阵 C 为实对称矩阵, 根据定理 2.25 可知其特征向量是正交的, 则以矩阵 C 的特征向量为行元素组成的矩阵就是要求的矩阵, 它可以将顶点集合中的顶点变换到使这些顶点坐标互不相关的空间中去。

这时, 求解顶点集合的自然轴问题就转换为计算协方差矩阵的特征向量的问题。为了计算协方差矩阵的特征向量, 必须首先计算协方差矩阵的特征值, 也就是该矩阵的特征多项式的根, 对于一个 3×3 的协方差矩阵其特征多项式为一个 3 次多项式。由于协方差矩阵为对称矩阵, 根据定理 2.24 知它的特征值为实数, 因此可以使用 5.1.2 一节中介绍的方法直接计算出它的特征值。通过求解相应的齐次线性方程组, 就可以获得与每个特征值对应的特征向量。

【例 7.1】 计算以下点集的自然轴:

$$P_1 = \langle -1, -2, 1 \rangle$$

$$P_2 = \langle 1, 0, 2 \rangle$$

$$P_3 = \langle 2, -1, 3 \rangle$$

$$P_4 = \langle 2, -1, 2 \rangle$$

解 首先计算平均位置 m 为

$$m = \frac{1}{4} \sum_{i=1}^4 P_i = \langle 1, -1, 2 \rangle \quad (7.5)$$

则协方差矩阵 C 为

$$C = \begin{bmatrix} \frac{3}{2} & \frac{1}{2} & \frac{3}{4} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{4} \\ \frac{3}{4} & \frac{1}{4} & \frac{1}{2} \end{bmatrix} \quad (7.6)$$

那么,该协方差矩阵的特征值就可以通过计算以下特征多项式的根而获得。

$$\det(C - \lambda I) = \begin{vmatrix} \frac{3}{2} - \lambda & \frac{1}{2} & \frac{3}{4} \\ \frac{1}{2} & \frac{1}{2} - \lambda & \frac{1}{4} \\ \frac{3}{4} & \frac{1}{4} & \frac{1}{2} - \lambda \end{vmatrix} = -\lambda^3 + \frac{5}{2}\lambda^2 - \frac{7}{8}\lambda + \frac{1}{16} \quad (7.7)$$

利用 5.1.2 一节中介绍的方法,可以直接获得特征多项式(7.7)的根,也就是协方差矩阵的特征值为

$$\begin{aligned} \lambda_1 &= 2.097 \\ \lambda_2 &= 0.3055 \\ \lambda_3 &= 0.09756 \end{aligned} \quad (7.8)$$

最后,通过求解线性方程组 $C - \lambda_i I = 0$,可以获得协方差矩阵的特征向量,分别用 R, S 和 T 表示。略去计算细节,矩阵 C 的单位特征向量为

$$R = \begin{bmatrix} -0.833 \\ -0.330 \\ -0.443 \end{bmatrix} \quad S = \begin{bmatrix} -0.257 \\ 0.941 \\ -0.218 \end{bmatrix} \quad T = \begin{bmatrix} 0.489 \\ -0.0675 \\ -0.870 \end{bmatrix} \quad (7.9)$$

即为点集 P_i 的 3 个自然轴。

在本节最后,用 R, S 和 T 表示顶点集合的自然轴,其中 R 与协方差矩阵的最大特征值对应,其方向为主坐标轴的方向, S 和 T 的方向分别表示与协方差矩阵的次大和最小特征值对应的坐标轴的方向,也就是说,如果 λ_1, λ_2 和 λ_3 分别是与特征向量 R, S 和 T 对应的特征值,则 $|\lambda_1| \geq |\lambda_2| \geq |\lambda_3|$ 。

7.1.2 构造边界盒

现在,给定一个三角网格的顶点位置 P_1, P_2, \dots, P_N ,可以确定与该对象的自然轴对应的三个方向 R, S 和 T 。为了构造该对象的边界盒,还需要计算顶点集沿这三个方向的最小与最大分布范围,从而获得边界盒的六个平面。其他类型的边界体还需要进一步的计算才能获得。

点集沿自然轴方向的分布范围可以通过简单地计算每个顶点向量与单位向量 R, S 和

T 之间的点乘,并通过取最小和最大值获得,则边界盒的6个面为

$$\begin{aligned} & \left\langle \mathbf{R}, -\min_{1 \leq i \leq N} \{P_i \cdot \mathbf{R}\} \right\rangle \quad \left\langle -\mathbf{R}, -\max_{1 \leq i \leq N} \{P_i \cdot \mathbf{R}\} \right\rangle \\ & \left\langle \mathbf{S}, -\min_{1 \leq i \leq N} \{P_i \cdot \mathbf{S}\} \right\rangle \quad \left\langle -\mathbf{S}, -\max_{1 \leq i \leq N} \{P_i \cdot \mathbf{S}\} \right\rangle \\ & \left\langle \mathbf{T}, -\min_{1 \leq i \leq N} \{P_i \cdot \mathbf{T}\} \right\rangle \quad \left\langle -\mathbf{T}, -\max_{1 \leq i \leq N} \{P_i \cdot \mathbf{T}\} \right\rangle \end{aligned} \quad (7.10)$$

【例 7.2】 计算与例 7.1 中给定的点集自然对齐的边界盒的6个面。

解 式 7.9 给出了该点集的自然轴,点集中的四个点分别与方向 \mathbf{R}, \mathbf{S} 和 \mathbf{T} 之间的点乘如下式所示:

$$\begin{cases} P_1 \cdot \mathbf{R} = 1.05 & P_1 \cdot \mathbf{S} = -1.84 & P_1 \cdot \mathbf{T} = -1.22 \\ P_2 \cdot \mathbf{R} = -1.72 & P_2 \cdot \mathbf{S} = -0.693 & P_2 \cdot \mathbf{T} = -1.25 \\ P_3 \cdot \mathbf{R} = -2.67 & P_3 \cdot \mathbf{S} = -2.11 & P_3 \cdot \mathbf{T} = -1.56 \\ P_4 \cdot \mathbf{R} = -2.22 & P_4 \cdot \mathbf{S} = -1.89 & P_4 \cdot \mathbf{T} = -0.695 \end{cases} \quad (7.11)$$

将 $P_1 \cdot \mathbf{R}$ 的最小与最大值带入式(7.10),得与方向 \mathbf{R} 垂直的两个平面

$$\langle \mathbf{R}, 2.67 \rangle \quad \langle -\mathbf{R}, 1.05 \rangle \quad (7.12)$$

同样,与方向 \mathbf{S} 和 \mathbf{T} 垂直的平面为

$$\begin{aligned} & \langle \mathbf{S}, 2.11 \rangle \quad \langle -\mathbf{S}, -0.693 \rangle \\ & \langle \mathbf{T}, 1.56 \rangle \quad \langle -\mathbf{T}, -0.695 \rangle \end{aligned} \quad (7.13)$$

边界盒的尺寸由 \mathbf{R}, \mathbf{S} 和 \mathbf{T} 方向上的最大与最小点积之间的差决定,在边界盒中心点 Q 处,经过两个反相面之间中线的三个平面相交。设三个标量系数 a, b 和 c 分别为点集在 \mathbf{R}, \mathbf{S} 和 \mathbf{T} 方向分布范围的平均值:

$$\begin{aligned} a &= \frac{\min_{1 \leq i \leq N} \{P_i \cdot \mathbf{R}\} + \max_{1 \leq i \leq N} \{P_i \cdot \mathbf{R}\}}{2} \\ b &= \frac{\min_{1 \leq i \leq N} \{P_i \cdot \mathbf{S}\} + \max_{1 \leq i \leq N} \{P_i \cdot \mathbf{S}\}}{2} \\ c &= \frac{\min_{1 \leq i \leq N} \{P_i \cdot \mathbf{T}\} + \max_{1 \leq i \leq N} \{P_i \cdot \mathbf{T}\}}{2} \end{aligned} \quad (7.14)$$

则经过两个反相面之间中线的三个平面就是可以分别将边界盒平分为平面 $\langle \mathbf{R}, -a \rangle, \langle \mathbf{S}, -b \rangle$ 和 $\langle \mathbf{T}, -c \rangle$,利用公式(4.20)可计算出这三个平面的交点,即边界盒的中心点 Q 为

$$Q = a\mathbf{R} + b\mathbf{S} + c\mathbf{T} \quad (7.15)$$

7.1.3 构造边界球

由于计算点集边界球的算法可以较快的速度执行,边界球通常被用于几何对象的可视性测试。与所有边界体一样,需要构造尽可能小的边界球,以使该边界球与视平截体相交的部分最小。获得一个绝对最优的边界球是相当困难的问题,这里不作进一步讨论,但是可以不进行复杂运算就可以获得一个有效的边界球,下面就介绍相应的算法。

为了计算给定点集 P_1, P_2, \dots, P_N 的边界球,首先要计算出主轴 \mathbf{R} 以及沿主轴方向上

与点集的最小和最大分布范围对应的点 P_k 和 P_l , 也就是点集中与 R 点积最小和最大的点, 然后构造一个球心位置为 Q 并且半径为 r 的球体

$$Q = \frac{P_k + P_l}{2} \quad (7.16)$$

$$r = \|P_k - Q\|$$

即球心位于主轴方向的点 P_k 和 P_l 的中线上, 半径为球心 Q 到 P_k 和 P_l 中任意一点的距离。

尽管式(7.16)给出的边界球接近于最终的边界球, 但是它不一定将点集 P_1, P_2, \dots, P_N 中的所有点包括在内, 因此必须对集合 $\{P_i\}$ 中的每个点进行测试, 以保证所有点被包含在边界球内部, 也就是根据边界球外部的点调整边界球的球心 Q 和半径 r , 并使所得新的边界球同时包含旧边界球和外部的点, 如图 7.3 所示, 对于点集中的一点 P_i , 如果

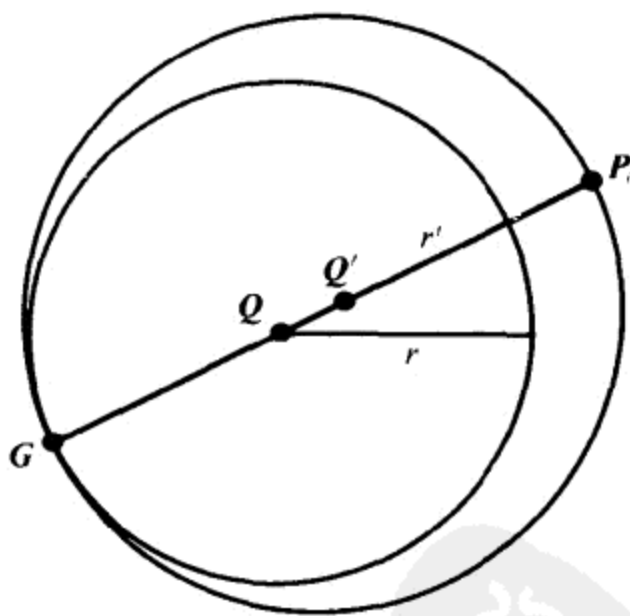
$$\|P_i - Q\|^2 > r^2 \quad (7.17)$$

则该点位于边界球之外。这时, 对当前的边界球进行以下调整以获得包含当前边界球和点 P_i 的新边界球。新边界球的球心位置 Q' 位于当前边界球球心 Q 与外部点 P_i 的连接线上, 由于新的边界球要与当前边界球相切, 切点为

$$G = Q - r \frac{P_i - Q}{\|P_i - Q\|} \quad (7.18)$$

该点也位于球心 Q 与外部点 P_i 的连接线上, 那么, 新的边界球的球心 Q' 为点 G 和 P_i 连线的中点, 其半径 r' 为新边界球球心与点 G 或者点 P_i 之间距离

$$Q' = \frac{G + P_i}{2} \quad r' = \|P_i - Q'\| \quad (7.19)$$



注:内部的边界球是由点集在主坐标轴方向上的分布范围决定的边界球,外部的边界球是包含所有点集中的点的扩展后的边界球

图 7.3 内部和外部的边界球

7.1.4 构造边界椭球

边界椭球适用于长条形的三角网格,为了确定顶点集合 P_1, P_2, \dots, P_N 的边界椭球,需要计算与点集自然轴 R, S 和 T 平行的椭球的三个半轴。该问题可以用以下方法解决,首

先,分别沿自然轴的方向对点集中的顶点位置进行一定比例的缩放变换,以使变换后点集的边界盒为一立方体,然后计算变换后点集的边界球,然后,对已经得到的边界球在自然轴方向上进行一定比例(大小为顶点位置缩放变换比例的倒数)的缩放,从而得出椭球三个半轴的长度。

为了对顶点位置进行缩放变换使点集的边界盒为立方体,需要知道在每个自然轴方向上与最小和最大分布对应的平面之间的距离,该距离等于标准边界盒的尺寸。也就是点 P_i 与向量 R, S 和 T 之间最大与最小点积之间的差,假设为 a, b, c ,它们的计算方法如下:

$$\begin{aligned} a &= \max_{1 \leq i \leq N} \{P_i \cdot R\} - \min_{1 \leq i \leq N} \{P_i \cdot R\} \\ b &= \max_{1 \leq i \leq N} \{P_i \cdot S\} - \min_{1 \leq i \leq N} \{P_i \cdot S\} \\ c &= \max_{1 \leq i \leq N} \{P_i \cdot T\} - \min_{1 \leq i \leq N} \{P_i \cdot T\} \end{aligned} \quad (7.20)$$

在 R 方向上对顶点位置进行比例为 $1/a$ 的缩放变换,同时分别在 S 和 T 的方向进行比例分别为 $1/b$ 和 $1/c$ 的缩放变换,就可以获得立方体边界盒。如方程(3.10)所示,执行缩放变换的变换矩阵 M 为

$$M = [R \ S \ T] \begin{bmatrix} 1/a & 0 & 0 \\ 0 & 1/b & 0 \\ 0 & 0 & 1/c \end{bmatrix} [R \ S \ T]^T \quad (7.21)$$

其中,由于向量 R, S 和 T 是正交的,用转置操作代替了求逆矩阵的操作。

对点集 $\{P_i\}$ 中的每个点进行缩放变换后,可以得到新的点集 MP_1, MP_2, \dots, MP_N 以及它的边界球。将该边界球的球心 Q 变换回原始点集的坐标空间就可以获得边界椭球的球心,可简单表示为 $M^{-1}Q$,其中,矩阵 M^{-1} 为矩阵 M 的逆矩阵

$$M^{-1} = [R \ S \ T] \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} [R \ S \ T]^T \quad (7.22)$$

利用点集 $\{MP_i\}$ 的边界球的半径 r 可以计算出边界椭球的三个半轴的长度,与自然轴方向 R, S 和 T 对应的半轴长度分别为 ar, br 和 cr 。

7.1.5 构造边界圆柱

圆柱形边界体由圆柱半径和其两端面圆心的坐标表示,圆柱的两个端面位于点集的边界盒中与主轴 R 垂直的平面上,是点集在主轴方向上的分布边界。求解一个点集的边界圆柱的过程中,大量的计算用于求一个边界圆,该圆包含点集中所有点在自然轴 S 和 T 确定的平面上的投影。

除了忽略点 P_i 中与主轴 R 平行的分量之外,边界圆的计算方法类似于边界球的计算。点集 $\{P_i\}$ 不能直接用于边界圆的计算,去掉点 P_i 在主轴 R 上投影后,可得新的点集 $\{H_i\}$ 为

$$H_i = P_i - (P_i \cdot R)R \quad (7.23)$$

由于自然轴 S 与协方差矩阵的第二大特征值对应,所以首先计算出点集 $\{H_i\}$ 中与向量 S 之间的点积为最小和最大的点 H_k 和 H_l ,则初始边界圆圆心 Q 和半径 r 分别为

$$Q = \frac{H_k + H_l}{2} \quad (7.24)$$

$$r = \|H_k - H_l\|$$

然后,与边界球的计算一样,对每个点进行包含性检查以确保它位于边界圆内。当一点 H_i 满足下述条件时:

$$\|H_i - Q\|^2 > r^2 \quad (7.25)$$

则需要对当前边界圆的圆心和半径进行以下调整:

$$Q' = \frac{G + H_i}{2} \quad (7.26)$$

$$r' = \|H_i - Q'\|$$

其中

$$G = Q - r \frac{H_i - Q}{\|H_i - Q\|} \quad (7.27)$$

点集 $\{H_i\}$ 边界圆的半径即为点集 $\{P_i\}$ 的边界圆柱的半径,边界圆的圆心 Q 位于与主轴 R 垂直但不一定经过原点的平面上,那么,将点 Q 向点集 $\{P_i\}$ 的边界盒中与主轴方向垂直的两个平面上投影,就可以得边界圆柱的两个端面的圆心 Q_1 和 Q_2 。

$$Q_1 = Q + \min_{1 \leq i \leq N} \{P_i \cdot R\} R \quad (7.28)$$

$$Q_2 = Q + \max_{1 \leq i \leq N} \{P_i \cdot R\} R$$

7.2 边界体测试

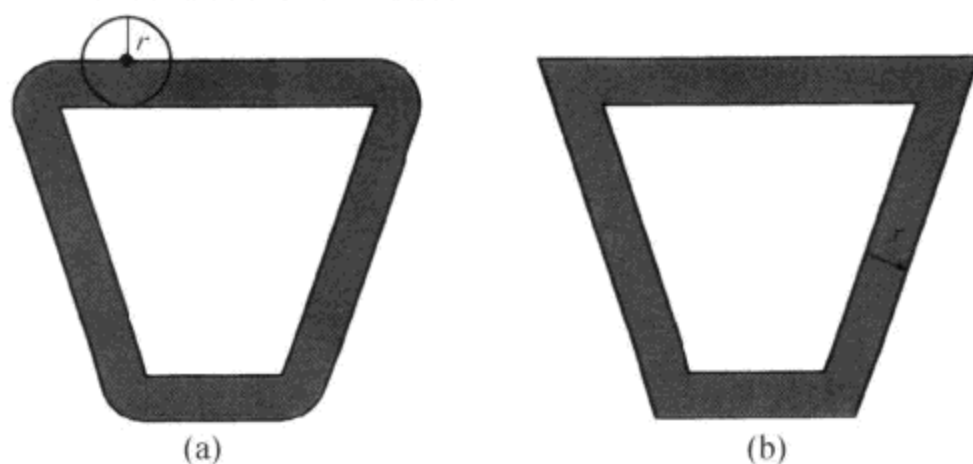
现在,已经知道各种不同边界体的构造方法,下面将介绍判断不同边界体的可视性的方法。本节中的技术将复杂的边界体与视平截体之间的求交运算转换为相对简单的点或者线段与一个经过适当修改的视平截体之间的求交运算。适当修改的视平截体是指根据边界体类型不同,对视平截体的面向外进行适当程度的移动。

7.2.1 边界球测试

如果半径为 r 的边界球的球心位于视平截体内部或者与视平截体的六个面中任意一个面之间的距离小于 r ,则该边界球与视平截体相交。如果一个边界球可见,则其球心必定位于图 7.4(a)中所示的灰色区域以内,该区域的边界是将边界球沿视平截体的外边滚动后形成的,除了在拐角处不与视平截体的面平行外,其他地方都与视平截体的面平行。图 7.4(b)中所示的灰色区域是将视平截体的六个面分别向外平移一段距离后形成的,是图 7.4(a)中所示的灰色区域的近似表示,这种表示可以简化边界球测试中的计算。

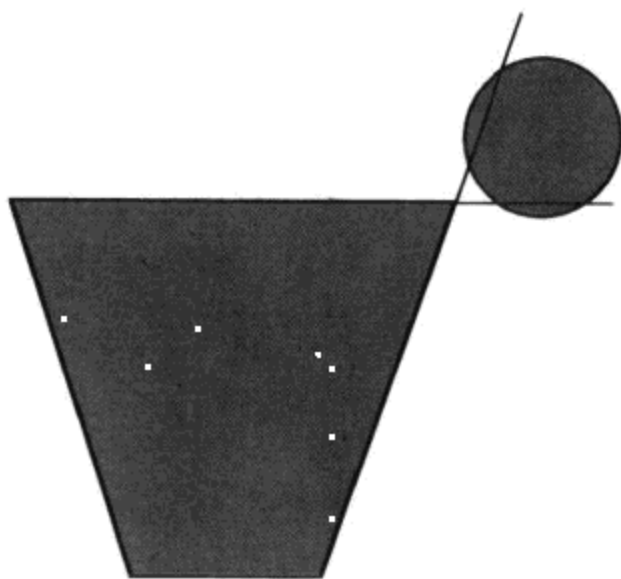
给定观察空间中球心位于点 Q 且半径为 r 的球体,计算点 Q 的齐次坐标与表 4.1 中所示的视平截体的六个面的 4D 点积,由于视平截体面的法线指向视平截体内部,负的点积表示点 Q 位于视平截体的外部,如果其中任何一个点积小于或者等于 $-r$,那么边界球与视平截体不相交,它所包含的对象就要被从可视几何体集合中去除,否则,边界球的一部分也许位于视平截体内部。有一种例外的情况,如图 7.5 所示,在视平截体边界附近,一些不可见

的边界球可能不会被从可视几何体集合中去除,这是由于这些边界球没有离视平截体的任一平面足够远而导致的。从简化可视性判断的计算来说,这种少数情况是可以允许的。在7.4.2一节中将介绍改善这种情况的方法。



注:(a)如果一个边界球可见,其球心必定位于灰色区域以内;(b)将视平截体的六个面分别向外平移距离 r 形成的灰色区域是原区域的近似表示

图 7.4 边界球与视平截体形成的灰色区域



注:在视平截体边界附近,一些边界球没有离视平截体的任一平面足够远而导致不可见的边界球被保留在可视几何体集合中

图 7.5 边界球的例外情况

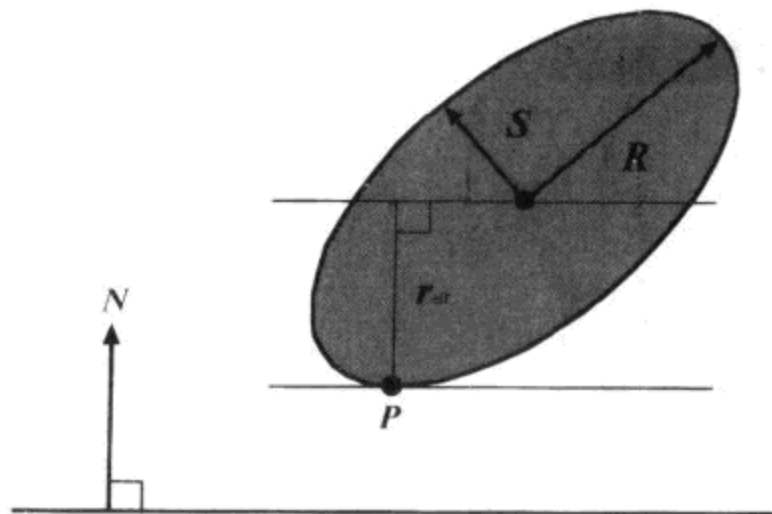
7.2.2 边界椭球测试

在测试球体的可视性时,视平截体的六个面被向外平移了边界球半径的距离,并测试边界球的球心是否位于移动后视平截体面的正区域中,也就是平面法线所指的区域中。类似的方法可以用于边界椭球的可视性判断中,但是,由于椭球不具备圆球那样的等轴对称性,它的有效半径对视平截体的各个平面来说是不相等的。

假设一个几何对象被半轴为互相垂直的向量 \mathbf{R} , \mathbf{S} 和 \mathbf{T} 的边界椭球包围,如图 7.6 所示,其中 \mathbf{R} , \mathbf{S} 和 \mathbf{T} 分别为平行于点集的自然轴,大小等于椭球半轴长的向量。椭球表面上一点 \mathbf{P} 可用 \mathbf{R} , \mathbf{S} 和 \mathbf{T} 表示为

$$\mathbf{P} = \mathbf{R} \sin\theta \cos\phi + \mathbf{S} \sin\theta \sin\phi + \mathbf{T} \cos\theta \quad (7.29)$$

上式是坐标轴与椭球轴对齐的球坐标系中的椭球表面点的表示方法,其中角 θ 为点 P 与向量 T 之间的夹角,角 ϕ 为点 P 在向量 R 和 S 确定的平面上的投影与向量 R 之间的夹角。对整个椭球面来说, θ 取值范围为 $0 \sim \pi$, ϕ 的取值范围为 $0 \sim 2\pi$ 。



注:半轴为互相垂直的向量 R, S 和 T 的边界椭球(T 指向页面的外面),其有效半径为椭球中心与椭球表面所有点向平面法线 N 的投影之间的最大距离

图 7.6 边界椭球及其有效半径

给定单位方向向量 N ,计算椭球面上向 N 的投影中数值最大的点 P ,可以获得椭球相对法向量为 N 的平面的有效半径 r_{eff} ,由于 N 为单位向量,点 P 到向量 N 投影的大小为 $P \cdot N$,为了获得点 P 到向量 N 的投影的大小为最大, θ 和 ϕ 的数值,设 $P \cdot N$ 相对于 θ 和 ϕ 的偏导数为 0,如下所示:

$$\frac{\partial}{\partial \theta}(P \cdot N) = (R \cdot N)\cos\theta\cos\phi + (S \cdot N)\cos\theta\sin\phi - (T \cdot N)\sin\theta = 0 \quad (7.30)$$

$$\frac{\partial}{\partial \phi}(P \cdot N) = -(R \cdot N)\sin\theta\sin\phi + (S \cdot N)\sin\theta\cos\phi = 0 \quad (7.31)$$

在 $P \cdot N$ 表达式的推导过程中,要使用以下三角恒等式:

$$\tan^2\alpha + 1 = \sec^2\alpha \quad (7.32)$$

它可以变换成以下两个三角恒等式

$$\sin\alpha = \frac{\tan\alpha}{\sqrt{\tan^2\alpha + 1}} \quad (7.33)$$

$$\cos\alpha = \frac{1}{\sqrt{\tan^2\alpha + 1}}$$

方程(7.31)可以变换为

$$(S \cdot N)\cos\phi = (R \cdot N)\sin\phi \quad (7.34)$$

则 $\tan\phi$ 可表示为

$$\tan\phi = \frac{S \cdot N}{R \cdot N} \quad (7.35)$$

方程(7.30)可以变换为

$$(T \cdot N)\sin\theta = (R \cdot N)\cos\theta\cos\phi + (S \cdot N)\cos\theta\sin\phi \quad (7.36)$$

则 $\tan\theta$ 可表示为

$$\begin{aligned}
\tan\theta &= \frac{\mathbf{R} \cdot \mathbf{N}}{\mathbf{T} \cdot \mathbf{N}} \cos\phi + \frac{\mathbf{S} \cdot \mathbf{N}}{\mathbf{T} \cdot \mathbf{N}} \sin\phi \\
&= \frac{1}{\sqrt{\tan^2\phi + 1}} \left(\frac{\mathbf{R} \cdot \mathbf{N}}{\mathbf{T} \cdot \mathbf{N}} + \tan\phi \frac{\mathbf{S} \cdot \mathbf{N}}{\mathbf{T} \cdot \mathbf{N}} \right) \\
&= \frac{\mathbf{R} \cdot \mathbf{N}}{\mathbf{T} \cdot \mathbf{N}} \frac{1}{\sqrt{\tan^2\phi + 1}} (1 + \tan^2\phi) \\
&= \frac{\mathbf{R} \cdot \mathbf{N}}{\mathbf{T} \cdot \mathbf{N}} \sqrt{\tan^2\phi + 1} \\
&= \frac{\mathbf{R} \cdot \mathbf{N}}{\mathbf{T} \cdot \mathbf{N}} \sqrt{\left(\frac{\mathbf{S} \cdot \mathbf{N}}{\mathbf{T} \cdot \mathbf{N}}\right)^2 + 1}
\end{aligned} \tag{7.37}$$

其中,最后一步中应用了三角恒等式(7.35)。利用三角恒等式(7.33)可得

$$\begin{aligned}
\mathbf{P} \cdot \mathbf{N} &= (\mathbf{R} \cdot \mathbf{N}) \sin\theta \cos\phi + (\mathbf{S} \cdot \mathbf{N}) \sin\theta \sin\phi + (\mathbf{T} \cdot \mathbf{N}) \cos\theta \\
&= \frac{1}{\sqrt{\tan^2\theta + 1}} \left\{ \frac{\tan\theta}{\sqrt{\tan^2\phi + 1}} [\mathbf{R} \cdot \mathbf{N} + (\mathbf{S} \cdot \mathbf{N}) \tan\phi] + \mathbf{T} \cdot \mathbf{N} \right\}
\end{aligned} \tag{7.38}$$

将式(7.35)和(7.37)代入式(7.38)得

$$\begin{aligned}
\mathbf{P} \cdot \mathbf{N} &= \frac{\frac{\mathbf{R} \cdot \mathbf{N}}{\mathbf{T} \cdot \mathbf{N}} \left(\mathbf{R} \cdot \mathbf{N} + \frac{(\mathbf{S} \cdot \mathbf{N})^2}{\mathbf{R} \cdot \mathbf{N}} \right) + \mathbf{T} \cdot \mathbf{N}}{\sqrt{\left(\frac{\mathbf{R} \cdot \mathbf{N}}{\mathbf{T} \cdot \mathbf{N}}\right)^2 \left[\left(\frac{\mathbf{S} \cdot \mathbf{N}}{\mathbf{R} \cdot \mathbf{N}}\right)^2 + 1 \right] + 1}} \\
&= \frac{(\mathbf{R} \cdot \mathbf{N})^2 + (\mathbf{S} \cdot \mathbf{N})^2 + (\mathbf{T} \cdot \mathbf{N})^2}{\mathbf{T} \cdot \mathbf{N} \sqrt{\left(\frac{\mathbf{R} \cdot \mathbf{N}}{\mathbf{T} \cdot \mathbf{N}}\right)^2 \left[\left(\frac{\mathbf{S} \cdot \mathbf{N}}{\mathbf{R} \cdot \mathbf{N}}\right)^2 + 1 \right] + 1}} \\
&= \frac{(\mathbf{R} \cdot \mathbf{N})^2 + (\mathbf{S} \cdot \mathbf{N})^2 + (\mathbf{T} \cdot \mathbf{N})^2}{\sqrt{(\mathbf{R} \cdot \mathbf{N})^2 + (\mathbf{S} \cdot \mathbf{N})^2 + (\mathbf{T} \cdot \mathbf{N})^2}}
\end{aligned} \tag{7.39}$$

最后,可得一个相对较简单的表达式

$$r_{\text{eff}} = \mathbf{P} \cdot \mathbf{N} = \sqrt{(\mathbf{R} \cdot \mathbf{N})^2 + (\mathbf{S} \cdot \mathbf{N})^2 + (\mathbf{T} \cdot \mathbf{N})^2} \tag{7.40}$$

通过式(7.41)可以计算出任意椭球体相对于单位法向量为 \mathbf{N} 的平面的有效半径,由于视平截体中最近和最远平面互相平行,所以椭球体相对于这两个平面的有效半径相等。这样,为了判断一个边界球是否位于视平截体之外,最多需要计算五个有效半径。与边界球测试一样,我们计算边界椭球的球心与视平截体各个面的法向量之间的 4D 点积,如果所有点积小于或者等于 $-r_{\text{eff}}$,则该边界椭球不可见,否则,该边界椭球包含的几何对象可见。

7.2.3 边界圆柱测试

球体和椭球体与视平截体的求交问题可以简化为测试一点是否位于平截体平面的正区域中(也就是平面法线所指的区域中)的问题,此处的平截体为将视平截体的面向外平移边界体有效半径的距离后形成的新平截体。为了简化边界圆柱与视平截体之间的求交问题,用一个线段在适当调整后的平截体中的可视性判断代替该问题的求解。

和边界椭球测试一样,在边界圆柱测试中,需要计算出边界圆柱相对于视平截体各个平面的有效半径。边界圆柱的有效半径依赖于圆柱的方向和区域,区域大小从 0(圆柱与平面

垂直时)到圆柱的实际半径(圆柱与平面平行时)。假设给定半径为 r 且端点为 Q_1 和 Q_2 的一个圆柱,设向量 \mathbf{A} 为与圆柱轴线平行的单位向量

$$\mathbf{A} = \frac{\mathbf{Q}_2 - \mathbf{Q}_1}{\|\mathbf{Q}_2 - \mathbf{Q}_1\|} \quad (7.41)$$

如图 7.7 所示,该圆柱相对于单位法向量为 \mathbf{N} 的平面的有效半径 r_{eff} 为

$$r_{\text{eff}} = r \sin \alpha \quad (7.42)$$

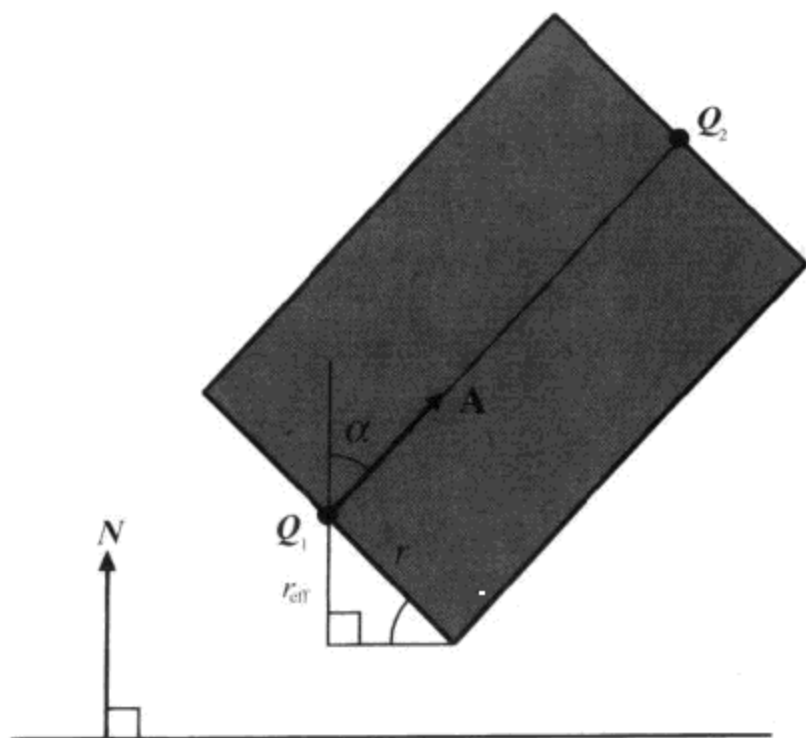


图 7.7 边界圆柱的有效半径

其中, α 为向量 \mathbf{A} 与 \mathbf{N} 之间的夹角,有效半径也可以表示为

$$r_{\text{eff}} = r \sqrt{1 - \cos^2 \alpha} = r \sqrt{1 - (\mathbf{A} \cdot \mathbf{N})^2} \quad (7.43)$$

由于视平截体中的最近与最远平面互相平行而使边界圆柱相对这两个平面的有效半径相同,在可视性判断时,首先处理最近与最远平面,然后再处理视平截体的其他平面。对于每个平截体的面 L ,首先计算圆柱端点 Q_1 、 Q_2 与它的 4D 点积 $L \cdot Q_1$ 和 $L \cdot Q_2$,如果二者小于或者等于 $-r_{\text{eff}}$ (与平面 L 对应的有效半径),则半径圆柱不可见,边界体测试停止。如果两个点积大于或者等于 $-r_{\text{eff}}$,则对其他平截体的面继续进行测试。

对于一个点积大于 $-r_{\text{eff}}$ 而另一个点积小于 $-r_{\text{eff}}$ 的情况,需要计算一点 Q_3 使

$$L \cdot Q_3 = -r_{\text{eff}} \quad (7.44)$$

然后,用它代替外部端点,这样可以有效地去除边界圆柱中已知位于视平截体外部的部分。用以下的直线参数方程可以计算出 Q_3 。

$$Q_3(t) = Q_1 + t(Q_2 - Q_1) \quad (7.45)$$

其中,为了表示圆柱的轴线 $0 \leq t \leq 1$ 。将式(7.45)的右边带入式(7.44),经变换可得 t 的计算公式

$$t = \frac{r_{\text{eff}} + L \cdot Q_1}{L \cdot (Q_2 - Q_1)} \quad (7.46)$$

其中, $Q_2 - Q_1$ 的差中含有一个值为 0 的 w 坐标。带入式(7.45)即得新的圆柱端点 Q_3 ,用

它替换圆柱的外部端点,然后继续处理剩下的视平截体平面。

如果处理完视平截体的六个面以后,从没有产生两个端点的点积都小于或者等于 $-r_{\text{eff}}$ 的情况,则边界圆柱至少部分可见。当然,对于最后处理的平面不必替换任何一个端点,只要遇到至少一个端点 Q_i 满足 $L \cdot Q_i > -r_{\text{eff}}$,则可判定边界圆柱与视平截体相交。

7.2.4 边界盒测试

在判断边界盒是否与视平截体相交时,可以将该问题简化为测试一点或者一条线段与适当调整过的视平截体之间的求交问题。当边界盒在主轴 R 的方向上的尺寸远远大于在 S 和 T 轴方向上的尺寸时,通常采用线段测试进行边界盒与视平截体的相交判断,而对于在三个轴方向上的尺寸近似相等时,则采用点测试方法。

这里,假设边界盒包围的几何对象的三个主轴向量 R, S 和 T 的大小等于边界盒本身的尺寸。为了将边界盒与视平截体之间的求交问题简化为判断边界盒的中心点是否位于适当放大后的视平截体的各个面内部的问题,需要计算出边界盒的有效半径。如图 7.8 所示,边界盒对于单位法向量为 N 的平面的有效半径为 r_{eff} 。

$$r_{\text{eff}} = \frac{1}{2}(|R \cdot N| + |S \cdot N| + |T \cdot N|) \quad (7.47)$$

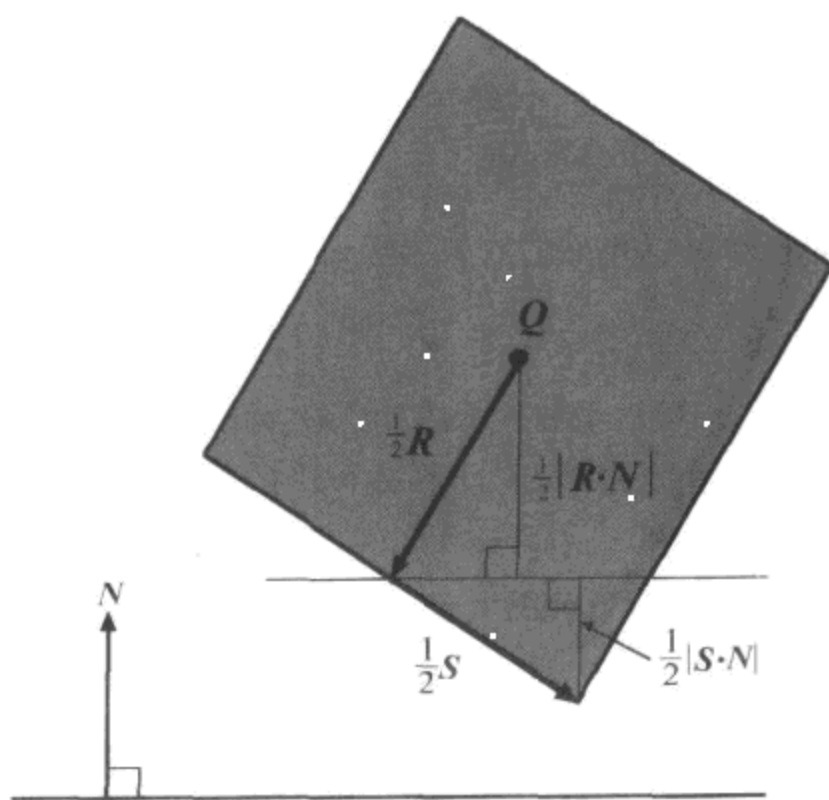


图 7.8 计算边界盒的有效半径

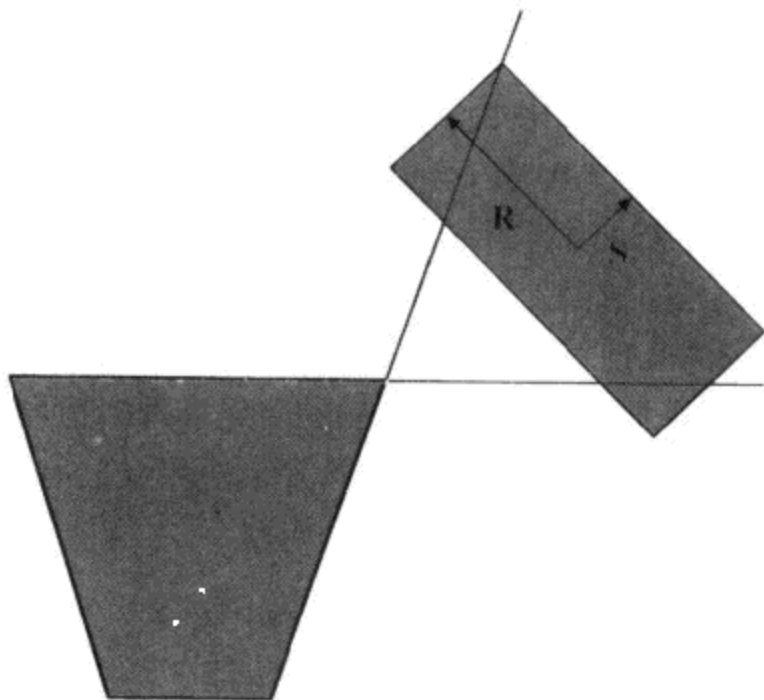
计算出该有效半径后,与椭球体测试中的方法一样,对于平截体中的每个平面 L ,计算边界盒中心点与该平面的 4D 点积,如果对于任何平面 $L \cdot Q < -r_{\text{eff}}$,则边界盒不可见。

当向量 R 的大小远远大于向量 S 和 T 的大小时,如果用点测试的方法,即使边界盒离平截体较远,也会得出相交的结果,图 7.9 中所示的就是这种情况的一个例子,为了解决这个问题,将边界盒测试简化为边界圆柱测试中的线段测试。

边界盒可以表示为一条线段,该线段的两个端点 Q_1 和 Q_2 可以表示成边界盒中心 Q

与其主轴 R 的表达式

$$\begin{aligned} Q_1 &= Q + \frac{1}{2}R \\ Q_2 &= Q - \frac{1}{2}R \end{aligned} \quad (7.48)$$



注:当边界盒主轴方向的尺寸远远大于其他两轴方向上的尺寸时,用点测试的方法会导致错误的判断

图 7.9 用点测试的方法导致错误判断的情况

边界盒相对于单位法向量为 N 的平面的有效半径 r_{eff} 为

$$r_{\text{eff}} = \frac{1}{2}(|S \cdot N| + |T \cdot N|) \quad (7.49)$$

其中,式(7.47)中的 $|R \cdot N|$ 一项由于被表示成端点 Q_1 和 Q_2 的线段而被省略。

接下来,与圆柱测试中的方法一样,对于平截体中的每个平面 L ,计算线段端点 Q_1 和 Q_2 与该平面的 4D 点积 $L \cdot Q_1$ 和 $L \cdot Q_2$,如果这两个点积都小于或者等于 $-r_{\text{eff}}$,则边界盒不可见,边界体测试停止。如果两个点积大于或者等于 $-r_{\text{eff}}$,则对其他平截体面继续进行测试。对于一个点积大于 $-r_{\text{eff}}$ 而另一个点积小于 $-r_{\text{eff}}$ 的情况,就按式(7.45)与(7.46)计算出一点 Q_3 ,使 $L \cdot Q_3 = -r_{\text{eff}}$,然后用它代替外部端点继续进行测试。如果处理完视平截体的六个面以后,从没有产生两个端点的点积都小于或者等于 $-r_{\text{eff}}$ 的情况,则边界圆柱至少部分可见。

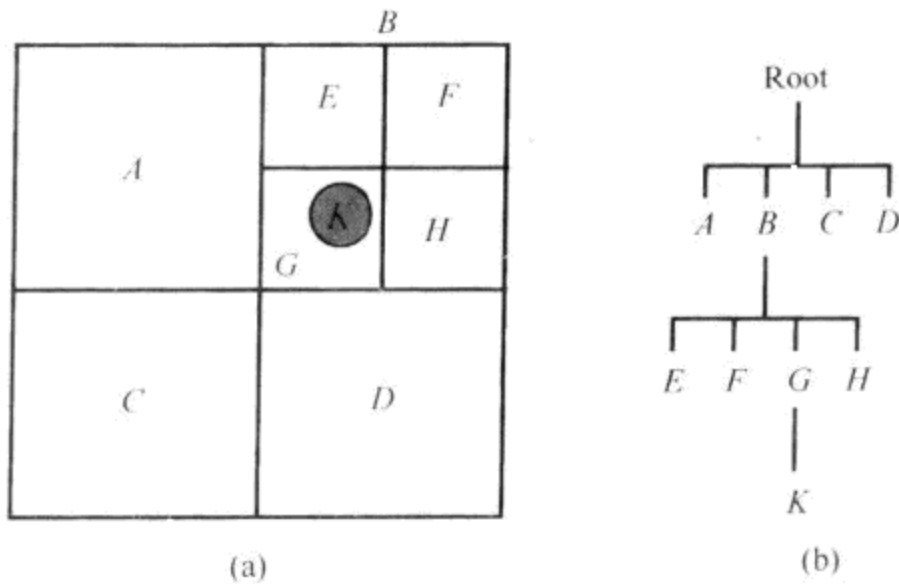
7.3 空间分割

将几何对象集合存储在一个结构中,并且利用该结构的特性,经过简单的测试就可以将大量的几何对象从可视集合中去除,这样就可以提高可视性判断的效率。实际上,已经存在这样的技术,即空间分割法,其中最常用的结构为八叉树和二叉空间分割树。这两种方法通常用于静态场景的计算,这是因为八叉树和二叉空间分割树这两种数据结构计算非常费时以致不能进行实时计算。

7.3.1 八叉树

假设整个场景或者场景的某个区域中的所有几何体可以包含在一个立方体盒子 B 中, 八叉树结构是指将该立方体盒子平均分割成八个等尺寸的小立方体, 这些小立方体又被进一步分割成八个更小尺寸的立方体, 该过程重复执行的次数称为八叉树的深度, 而该过程执行过程中产生的立方体成为八分体, 每个八分体被连接到生成它的立方体盒子中, 场景中的每个几何对象被连接到完全包含它的八分体中(也可能是原始立方体盒 B 中)。

图 7.10(a)表示的是八叉树的二维模拟, 称为四叉树, 图中的四叉树为只包含一个对象的区域的四叉树。图 7.10(b)则说明了该四叉树结构是如何组织起来的, 四叉树中的每个节点最多有四个子节点, 同样, 八叉树中的节点最多有八个子节点。如图 7.10 中的例子所示, 如果场景中没有几何对象与某个四分体(对八叉树来说为八分体)相交, 则该四分体停止进一步细分。并且, 不完全包含任何几何对象的四分体将被从树结构中删除, 因此, 任何不存在的四分体都被认为是空的。



注:(a)只包含一个对象的四叉树;(b)四叉树的数据结构

图 7.10 只包含一个对象的区域的四叉树及其数据结构表示

通过将场景中的几何体组织到八叉树中, 只要知道树中的某个节点不可见, 则它的所有子节点也不可见, 从而可以被从可视集合中删除。第 8 章中还会看到这种数据结构对碰撞检测的作用。利用八叉树进行可视性判断时, 首先对包含根节点的立方体盒与视平截面之间进行求交运算。如果已知观察位置位于八叉树的边界内, 则根节点总是可见的。已知包含某个节点的立方体盒可见, 则对连接到该节点的几何对象的边界体进行相应的可视性判断。然后再对该可视节点的子节点进行相同的可视性测试。如果一个节点在边界盒测试后为不可见节点, 则所有连接到该节点的几何对象和该节点的子节点都不可见。

事实上, 在八叉树的每一层的八分体对应的边界盒方向相同, 因此, 根据给定的观察位置与方向, 将八叉树主轴变换到观察空间并计算包含整个八叉树的立方体 B 的五个有效半径, 其中一个有效半径是相对视平截体的最近与最远面, 四个为相对其他四个视平截面面的有效半径。如果 r_{eff} 为包含整个八叉树的立方体盒 B 相对一个视平截面面的有效半径, 则位于八叉树第一层的立方体盒相对于该平面的有效半径为 $r_{\text{eff}}/2$ 。这样, 就不必应用式 (7.47) 去计算八叉树中每层每一八分体的有效半径, 而只需在开始时计算一次就可以。

7.3.2 二叉空间分割树

二叉空间分割树简称 BSP 树,它是一种在每层将某个空间分割成两部分的结构。与八叉树不一样,BSP 树的分割平面可以是任意方向。一个几何对象集合的 BSP 树是通过选择分割平面(也称分离面)将几何对象集合中的几何体分成两组而形成的,其中一组几何体位于分割平面的正向区域(正半空间),另外一组则位于负向区域(负半空间)。

传统上,BSP 树的分割平面与几何体的多边形面对齐,图 7.11 中所示的是一个二维的例子,其中 BSP 树的结构由包含多个多边形的区域决定。BSP 树的每层选择一个多边形所在平面作为分割平面,其余的多边形被划分到正半空间和负半空间中,与分割平面相交的多边形则被分割成分别位于正半空间和负半空间的两个多边形,正半空间和负半空间中的多边形被进一步分割,直到处理完所有多边形。

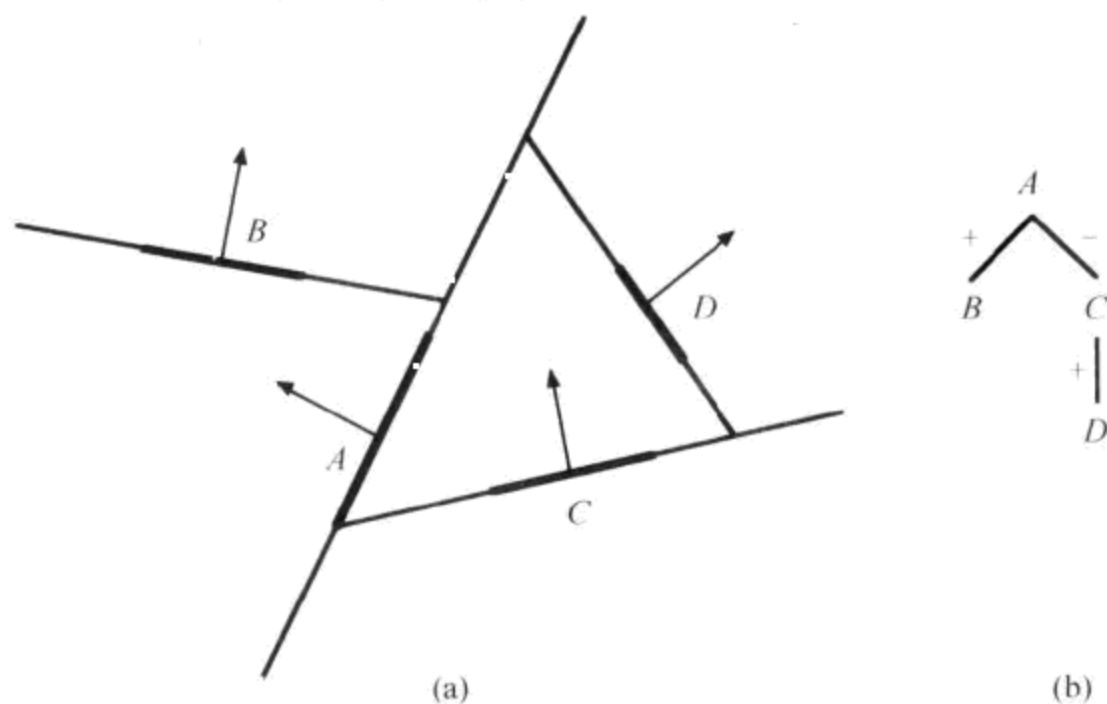


图 7.11 传统的 BSP 树及其数据结构表示

大量多边形和曲面在现代 3D 引擎中的应用使传统 BSP 树不适用实际问题的解决。为了使用 BSP 树,需要进行适当修改,即为每个几何对象构造新的分割面而不是采用几何对象的多边形面为分割面。如图 7.12 所示,几何对象的分割面与几何对象的主轴 T 垂直,而 T 为与几何对象的边界盒的最小边长对应的主轴,这样可使几何对象与分割平面之间的距离最小。确定几何对象的分割面后,其他对象就可划分为完全位于正半空间和完全位于负半空间中的两组,而跨越分割面的对象被同时放在正半空间和负半空间内。半空间被继续分割直到处理完所有几何对象。

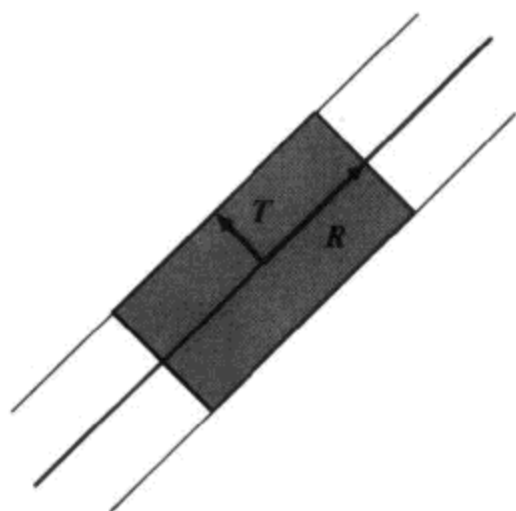
对于 BSP 树中的每个分割面,需要计算每个半空间可视性和该分割面所在几何对象的可视性,这需要进行平面 K 与视平截体之间的求交运算。最简单的方法是计算出视平截体的八个顶点与平面 K 之间的点积,然后将这些点积与数值 0 进行比较,如果八个点积具有相同的符号,则说明相应的八个顶点位于平面 K 的同一侧,那么平面与视平截体不相交。还可以使用一个更好的方法进行平面与视平截体之间的求交运算,该方法将平面 K 变换到齐次裁剪空间,同时利用了视平截体在该空间的三次对称性(见 4.5.1 一节)。

利用以下公式,可以将平面 K 从世界坐标空间变换到齐次裁剪空间

$$\mathbf{K}' = [(\mathbf{PM})^{-1}]^T \mathbf{K}$$

其中, \mathbf{P} 为投影矩阵, \mathbf{M} 为从世界坐标空间到观察空间的变换矩阵。裁剪空间中视平截体的顶点坐标值的取值范围为 ± 1 。与平面 \mathbf{K}' 的点积最大的顶点为坐标值符号与向量 \mathbf{K}' 的分量符号一致的顶点, 相反, 与平面 \mathbf{K}' 的点积最小的顶点为坐标值符号与向量 \mathbf{K}' 的分量符号相反的顶点。最大点积 d_{\max} 与最小点积 d_{\min} 如下式所示:

$$\begin{aligned} d_{\max} &= |K'_x| + |K'_y| + |K'_z| + K'_w \\ d_{\min} &= -|K'_x| - |K'_y| - |K'_z| + K'_w \end{aligned} \quad (7.51)$$



注: T 为与几何对象边界盒的最小边长对应的主轴
图 7.12 几何对象的分割平面与主轴 T 垂直

如图 7.13 所示, 如果 $d_{\max} \leq 0$, 则视平截体完全位于平面 \mathbf{K} 的负半空间, 这意味着在平面 \mathbf{K} 的正半空间中的几何对象都不可见。类似, 若 $d_{\min} \geq 0$, 则视平截体完全位于平面 \mathbf{K} 的正半空间, 这意味着在平面 \mathbf{K} 的负半空间中的几何对象都不可见。如果条件 $d_{\max} \leq 0$ 和 $d_{\min} \geq 0$ 都不成立, 则平面 \mathbf{K} 与视平截体相交, 任何一个半空间都不能被去除。

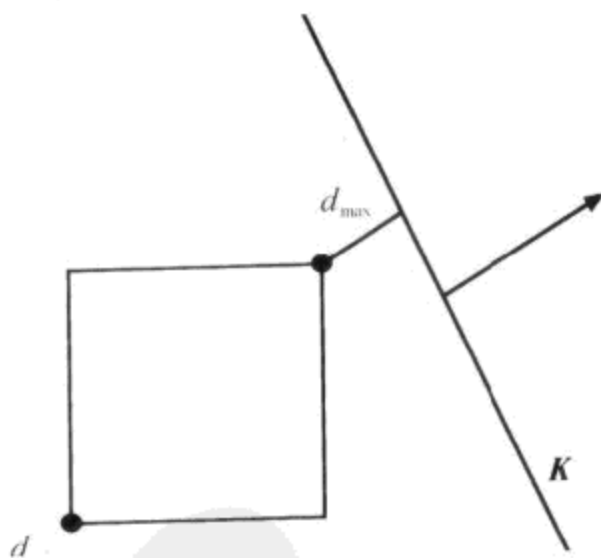


图 7.13 d_{\max} 和 d_{\min} 为视平截体所有顶点与平面 \mathbf{K} 的最大和最小点积

7.4 门系统

门系统是可视性判断中非常有效的一种技术, 它可以快速地从进行可视性判断的场景

中去除大量的不可见区域。它来自一个非常简单的想法,即场景可被分割成多个不相连的区间,这些区间可以通过门联系起来。门通常用凸多边形表示,通过它,一个区间可以被另一个区间看到。门系统的优点是如果任何区间不能通过一系列的门看到,则该区间就不可见而不必进行显示渲染。在用门系统判断一个场景的某一部分是否可见时,可能只需要处理整个数据集合的一小部分,这是因为任何位于不可见门的正半空间中的几何体不参加运算。

图 7.14 中说明了如何用门系统进行可视性判断。首先确定观察点位于的区间,该区间总是被认为是可见的,然后对离开该区间的每个门进行测试,对于与视平截面相交的门,其对应的区间被认为是可见的。每个离开关联区间的门(不包括引进第一个区间的门)都要进行可视性判断,不过可视性判断中的视平截体的尺寸已经被正观察的门的边界缩小了。门系统方法被重复执行直到没有新门是可见的。

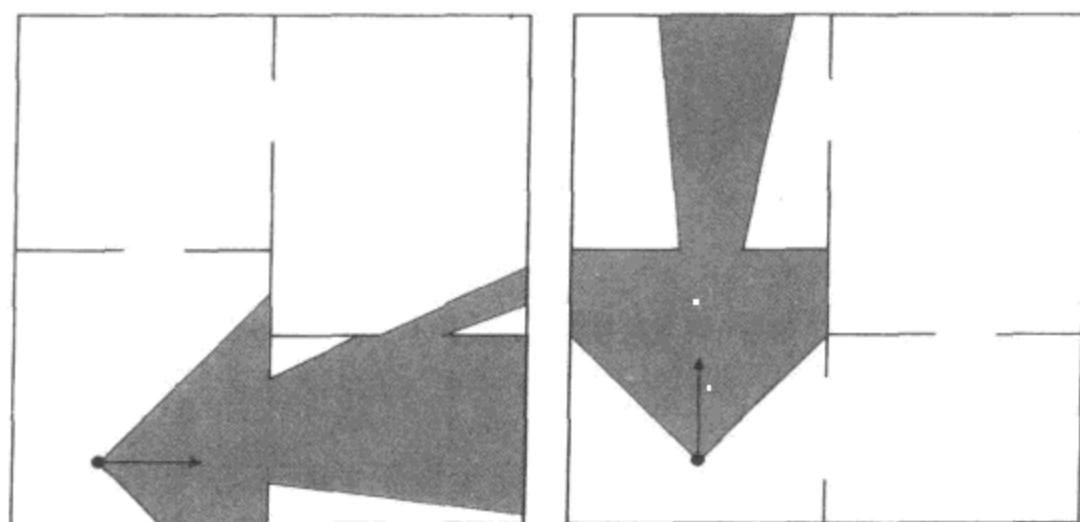


图 7.14 只有那些通过一系列的门可以看到的区间被认为是可见的

门所联系的区间可以用八叉树或者 BSP 树表示,在这些区域中的几何对象仍然有自己的边界体。用门系统对大区间进行的可视性判断要进行大规模裁剪处理,对其产生的小区间还要进行小规模的可视性判断以完成整个区间的可视性检查。

7.4.1 门裁剪

从观察点通过一个区间的门观察该区间时,该区间中的可视体要小于整个视平截面。这样,在与新的较小的视平截面进行小规模可视性测试时,要处理对象的数量就会大大减少。初始视平截面中的最近与最远面保持不变,其他的面则被一组新面替代,从而形成新的视平截面,这组新面是初始视平截面与表示门的多边形的交面。

约定包含门的平面的法线方向指向观察点,门的顶点按逆时针方向排列,如图 7.15 所示。因此,场景中的门是单向的,即如果一个门为从区间 X 到区间 Y 的门,则通过该门就不能从区间 Y 向区间 X 观察。当观察点位于包含门的平面的负半空间中时,该门被认为是不可见的。双向可见的两个

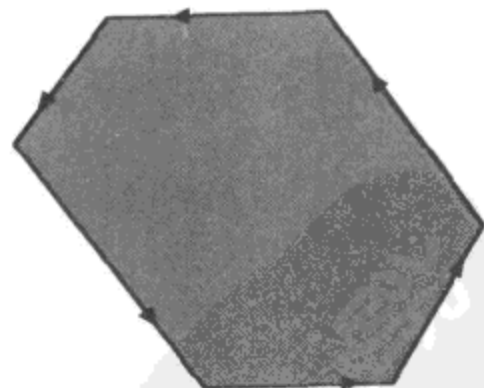


图 7.15 门的顶点相对于包含它们的平面法线按逆时针排列

区间需要每个区间中都包含一个可以看到另一区间的门。

当处理观察某个区间的门时,应只对门的可见区域感兴趣。门的可见区域等于门与视平截面相交的部分,它可能为初始的视平截面,也可能为尺寸缩小了的一个视平截面。为了确定门的可见区间,用当前视平截体的面裁剪表示一个门的多边形。多边形被一个平面裁剪后,它位于裁剪面的负半空间中的部分将被去除,剩余的部分为一个完全位于裁剪面的正半空间中的新多边形。用视平截体的所有面对门的多边形裁剪后,多边形位于可见空间外的部分就被有效地去除了。

假设要裁剪一个门,该门为顶点 V_1, V_2, \dots, V_n 依次连接形成的凸多边形,用一个平面 L 对该多边形裁剪后,将获得一个最多有 $n+1$ 个顶点的凸多边形。在进行门裁剪前,将门的顶点分成三类,即位于平面 L 的正半空间中的顶点、位于平面 L 负半空间中的顶点和位于平面 L 上的顶点。其中,如果一个顶点 V_i 位于平面 L 上,则该顶点满足以下条件:

$$-\epsilon < L \cdot V_i \leq 0 \quad (7.52)$$

其中, ϵ 为某一任意小的常数,如 $\epsilon = 0.001$ 。这样可以避免由于舍入误差造成的对视平截体的凸性破坏,从而引起的对可视性计算的崩溃。如果没有顶点位于平面 L 的正半空间,则门是不可见的,不显示它所关联的区间的任何几何对象。如果没有顶点位于平面 L 的负半空间,则不对门进行裁剪。否则,将遍历门的所有相邻顶点,寻找一个顶点位于平面 L 的正半空间,而另一顶点位于负半空间中的门的多边形边,如图 7.16 所示,边与裁剪平面 L 的交点成为多边形的新顶点,而位于负半空间中的顶点则被去除了。位于裁剪面正半空间和平面 L 上的顶点保持不变,仍为新多边形的顶点。

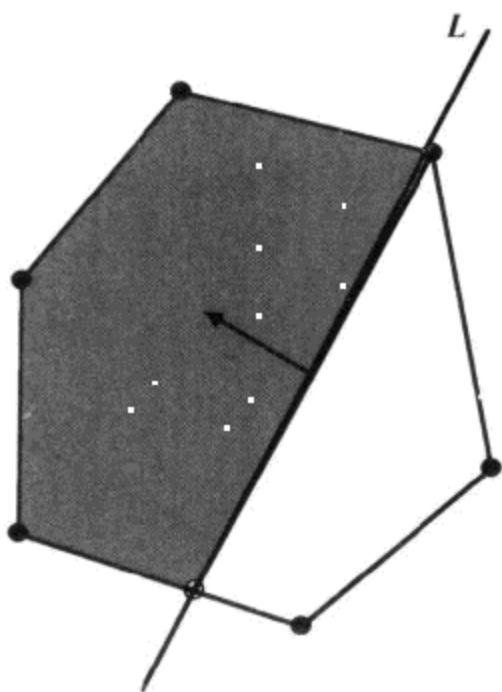


图 7.16 多边形边与裁剪平面的交点成为新多边形的顶点

假设顶点 V_i 位于裁剪平面 L 的正半空间,顶点 V_{i+1} 位于 L 的负半空间,即顶点 V_i 和 V_{i+1} 满足以下条件:

$$\begin{aligned} L \cdot V_i &> 0 \\ L \cdot V_{i+1} &< -\epsilon \end{aligned} \quad (7.53)$$

线段 $V_i V_{i+1}$ 上的一点 W 为

$$W(t) = V_i + t(V_{i+1} - V_i) \quad (7.54)$$

其中,参数 t 的取值范围为 $0 \leq t \leq 1$ 。满足 $L \cdot W(t) = 0$ 的 t 值为

$$t = \frac{L \cdot V_i}{L \cdot (V_i - V_{i+1})} \quad (7.55)$$

其中, $V_i - V_{i+1}$ 的差中含有一个为 0 的 w -坐标。将式(7.55)代入式(7.54)可得新的顶点 W 。

7.4.2 缩小视平载体

给定一个裁剪后的门,希望得到通过该门的可见空间的边界平面,从而在可视性判断时,采用新的缩小了的视平载体,这样,大量的几何对象就会不参加运算。幸运的是,在观察空间中,可以很容易地得到与门的边对应的平面。设平面 L_i 为经过原点与门中一条边的两个顶点 V_i 和 V_{i+1} 的平面,其计算公式为

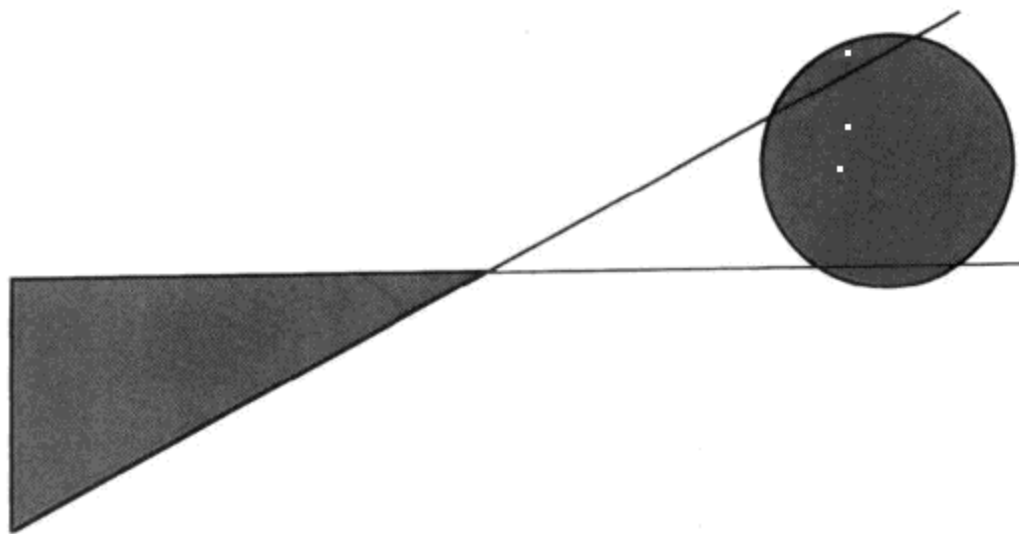
$$L_i = \left\langle \frac{V_{i+1} \times V_i}{\|V_{i+1} \times V_i\|}, 0 \right\rangle \quad (7.56)$$

对于含有 n 个顶点的门,要计算 n 个面,这 n 个面为缩小后的视平截体的面,对于第 n 个面 $V_{n+1} = V_0$ 。如果相邻的门顶点之间的距离非常小,舍入误差将导致视平截体的凸性变化,为了防止该问题,将去除满足以下条件的平面 V_i :

$$\|V_{i+1} - V_i\|^2 < \epsilon$$

其中, ϵ 为一无穷小的常数,通过调整它的值,可以避免上述平面的产生。

缩小后的视平截体的面之间的夹角可能非常小,如图 7.17 所示,这将影响可视性判断的有效性,这是因为即使一个对象离视平载体很远,它也不可能完全位于某些视平载体平面的负半空间。通过为夹角非常小的两个视平载体面之间附加一个新平面,可以避免这种问题。



注:图中边界球离视平载体很远,但仍然得出可见的判断结果

图 7.17 缩小后的视平载体的相邻面之间非常小的夹角

图 7.18 所示的就是一个法向量为 N_3 的新平面被加到视平载体中,该平面为法向量分别为 N_1 和 N_2 的相邻视平载体面之间的平面。向量 N_3 为向量 N_1 和 N_2 之间的平均(非

规则化)方向,即为 $N_1 + N_2$,然后减去该向量在 $N_1 \times N_2$ 方向上的投影就可以得到新的平面 N_3 ,这样做是为了保证新平面包含平面 N_1 和 N_2 的交线。新平面的表达式如下:

$$\begin{aligned} A &= N_1 + N_2 \\ B &= N_1 \times N_2 \\ N_3 &= \frac{A - (A \cdot B)B}{\|A - (A \cdot B)B\|} \end{aligned} \quad (7.58)$$

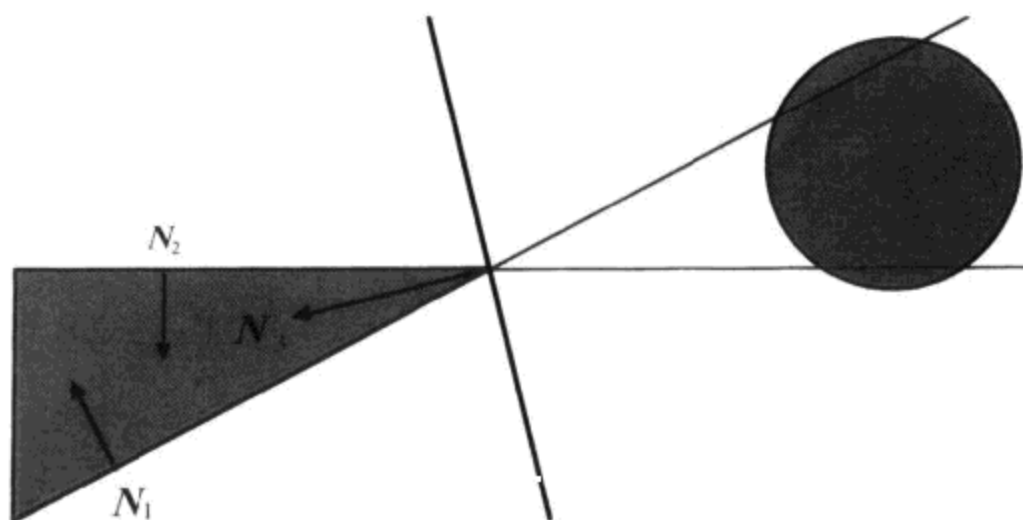


图 7.18 增加一个新平面可保证可视性判断正确执行

由于新平面经过观察空间的原点,则它的 w -坐标为 0。当两个法向量分别为 N_1 和 N_2 的相邻视平截面平面满足 $N_1 \cdot N_2 < \alpha$ (α 为小角度门限)条件时,就会遇到图 7.17 中所示的情况,这种情况可以通过增加一个新平面而避免。由于新加的平面包含两个相邻视平截体面的交线,因此它并不能成为视平截体的面,它仅用于某一个区间的可视性测试而并不参加对其他区间的门的裁剪操作。

7.5 本章小结

主分量

含有 N 个顶点 P_1, P_2, \dots, P_N 的点集的主轴 R, S 和 T 是协方差矩阵 C 的特征向量,协方差矩阵 C 为

$$C = \frac{1}{N} \sum_{i=1}^N (P_i - m)(P_i - m)^T$$

其中 m 为平均位置

$$m = \frac{1}{N} \sum_{i=1}^N P_i \quad (7.5)$$

如果 λ_1, λ_2 和 λ_3 分别为与特征向量 R, S 和 T 对应的特征值,则 $|\lambda_1| \geq |\lambda_2| \geq |\lambda_3|$ 。

边界盒

与主轴 A 垂直并为点集 P_1, P_2, \dots, P_N 在该轴方向上的边界的两个平面,其表达式为

$$\left\langle \mathbf{A}, -\min_{1 \leq i \leq N} \{ \mathbf{P}_i \cdot \mathbf{R} \} \right\rangle \quad \left\langle -\mathbf{A}, -\max_{1 \leq i \leq N} \{ \mathbf{P}_i \cdot \mathbf{R} \} \right\rangle$$

边界盒的中心 Q 为

$$Q = k_1 \mathbf{A}_1 + k_2 \mathbf{A}_2 + k_3 \mathbf{A}_3$$

其中

$$k_j = \frac{\min_{1 \leq i \leq N} \{ \mathbf{P}_i \cdot \mathbf{A}_j \} + \max_{1 \leq i \leq N} \{ \mathbf{P}_i \cdot \mathbf{A}_j \}}{2}$$

$\mathbf{A}_1, \mathbf{A}_2$ 和 \mathbf{A}_3 为单位长度的主轴。

如果用向量 \mathbf{R}, \mathbf{S} 和 \mathbf{T} 表示边界盒的尺寸与方向, 则该边界盒相对于法向量为 \mathbf{N} 的平面的有效半径 r_{eff} 为

$$r_{\text{eff}} = \frac{1}{2} (|\mathbf{R} \cdot \mathbf{N}| + |\mathbf{S} \cdot \mathbf{N}| + |\mathbf{T} \cdot \mathbf{N}|)$$

边界球

在点集 $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_N$ 的边界球计算中, 首先确定与主轴 \mathbf{R} 的点积最小和最大的点 \mathbf{P}_k 和 \mathbf{P}_l , 并将边界球的初始球心和半径设为

$$Q = \frac{\mathbf{P}_k + \mathbf{P}_l}{2}$$

$$r = \|\mathbf{P}_k - Q\|$$

对于满足条件 $\|\mathbf{P}_i - Q\|^2 > r^2$ 的顶点 \mathbf{P}_i , 将球心位置和半径替换为以下数值:

$$Q' = \frac{G + \mathbf{P}_i}{2}$$

$$r' = \|\mathbf{P}_i - Q'\|$$

其中, G 的表达式为

$$G = Q - r \frac{\mathbf{P}_i - Q}{\|\mathbf{P}_i - Q\|}$$

如果相对视平截体的任一平面 L , 球心为 Q , 半径为 r 的边界球满足 $L \cdot Q \leq -r$, 则该边界球不可见。

边界椭球

在点集 $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_N$ 的边界椭球计算中, 首先将点集中的顶点变换到另一空间中, 在该空间中点集的边界盒为一立方体, 构造该空间中点集的边界球, 然后进行逆变换将边界球变换回原空间中边界盒的尺寸。

如果用向量 \mathbf{R}, \mathbf{S} 和 \mathbf{T} 表示边界椭球半轴的尺寸与方向, 则该边界椭球相对于法向量为 \mathbf{N} 的平面的有效半径 r_{eff} 为

$$r_{\text{eff}} = \sqrt{(\mathbf{R} \cdot \mathbf{N})^2 + (\mathbf{S} \cdot \mathbf{N})^2 + (\mathbf{T} \cdot \mathbf{N})^2}$$

如果相对视平截体的任一平面 L , 球心为 Q 的边界椭球满足 $L \cdot Q \leq -r_{\text{eff}}$, 则该边界球不可见。

边界圆柱

计算点集 P_1, P_2, \dots, P_N 的边界圆柱时,首先要用以下公式计算出新的点集 $\{H_i\}$:

$$H_i = P_i - (P_i \cdot R)R$$

其中, R 为与主轴平行的单位向量。计算出点集 $\{H_i\}$ 的边界圆,该圆的圆心为 Q ,半径为 r ,然后,计算边界圆柱轴线的两个端点 Q_1 和 Q_2 分别为

$$Q_1 = Q + \min_{1 \leq i \leq N} \{P_i \cdot R\} R$$

$$Q_2 = Q + \max_{1 \leq i \leq N} \{P_i \cdot R\} R$$

相对任一法向量为 N 的平面,边界圆柱的有效半径为 r_{eff}

$$r_{\text{eff}} = r \sqrt{1 - (A \cdot N)^2}$$

其中, A 为与圆柱轴线平行的单位向量

$$A = \frac{Q_2 - Q_1}{\|Q_2 - Q_1\|}$$

如果连接圆柱轴线的两个端点 Q_1 和 Q_2 的线段被视平截面平面完全裁剪,则边界圆柱不可见。

二叉空间分割树(BSP 树)

用以下公式将世界坐标系中的平面 K 变换到齐次裁剪空间,就可以判断出该平面与视平截面间的相交情况

$$K' = [(PM)^{-1}]^T K$$

其中, P 为投影矩阵, M 为从世界坐标空间到观察空间的变换矩阵。视平截体的顶点与平面 K' 最大点积 d_{max} 与最小点积 d_{min} 为

$$d_{\text{max}} = |K'_x| + |K'_y| + |K'_z| + K'_w$$

$$d_{\text{min}} = -|K'_x| - |K'_y| - |K'_z| + K'_w$$

如果 $d_{\text{max}} \leq 0$ 或者 $d_{\text{min}} \geq 0$,则视平截面完全位于平面 K 的一个半空间中,而另一个半空间是不可见的。

门系统

当用一个平面 L 裁剪由顶点 V_1, V_2, \dots, V_N 组成的门时,在位于平面 L 两侧的相邻顶点之间要增加新的顶点,该顶点用 W 表示为

$$W = V_i + t(V_{i+1} - V_i)$$

其中,参数 t 为

$$t = \frac{L \cdot V_i}{L \cdot (V_i - V_{i+1})}$$

通过原点与顶点 V_i 和 V_{i+1} 的平面为

$$L_i = \left\langle \frac{V_{i+1} \times V_i}{\|V_{i+1} \times V_i\|}, 0 \right\rangle \quad (7.56)$$

当相邻的两个法向量分别为 N_1 和 N_2 的两个平面之间夹角非常小时,为了提高边界体可视性测试的稳定性,要在视平截体中引入一个新平面。法向量为 N_3 并通过原点的新平面的定义为

$$\begin{aligned} \mathbf{A} &= \mathbf{N}_1 + \mathbf{N}_2 \\ \mathbf{B} &= \mathbf{N}_1 \times \mathbf{N}_2 \\ \mathbf{N}_3 &= \frac{\mathbf{A} - (\mathbf{A} \cdot \mathbf{B})\mathbf{B}}{\|\mathbf{A} - (\mathbf{A} \cdot \mathbf{B})\mathbf{B}\|} \end{aligned} \quad (7.58)$$

习 题

1. 给定球心分别为 Q_1 和 Q_2 , 半径分别为 r_1 和 r_2 的两个球体 S_1 和 S_2 , 求包含这两个球的最小的球体的球心 Q 与半径 r 。分别考虑两个球体 S_1 和 S_2 互相分离、相交以及包含的情况。
2. 计算图 7.19 中所示的底面半径为 s , 高为 h , 且底面中心位于 $x-y$ 平面原点的圆锥的最小边界球的球心 Q 与半径 r 。分别考虑 $s < h$ 和 $s \geq h$ 两种情况。

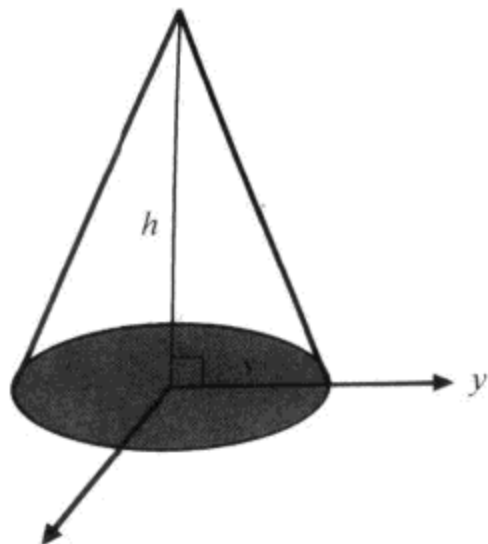
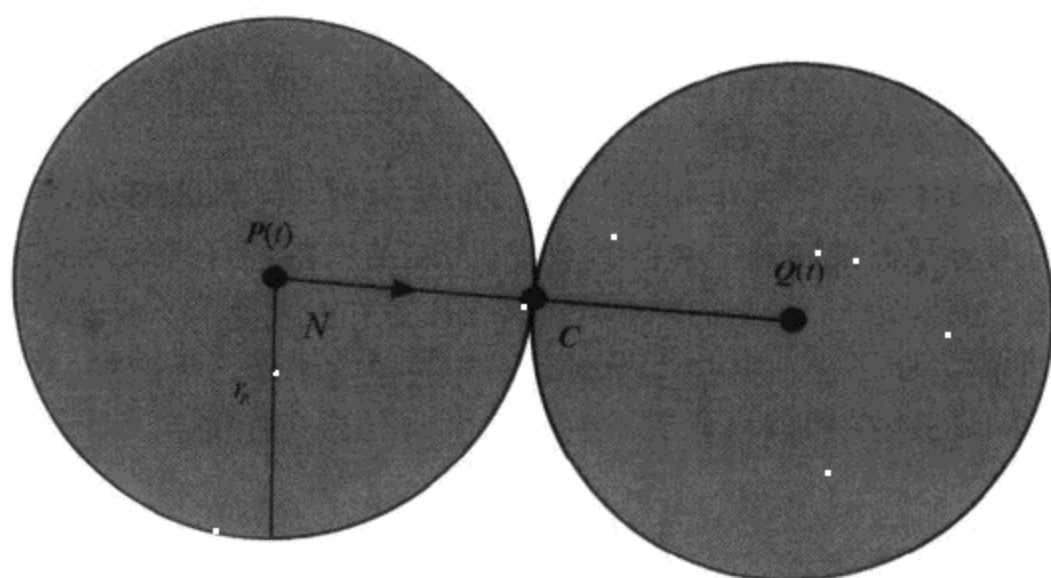


图 7.19 第 2 题中的圆锥

3. 计算边分别为 $\mathbf{R} = \langle 2, 0, 1 \rangle$, $\mathbf{S} = \langle 1, 0, -2 \rangle$ 和 $\mathbf{T} = \langle 0, 1, 0 \rangle$ 的边界盒相对于单位法向量为 $\mathbf{N} = \left\langle \frac{\sqrt{3}}{3}, -\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3} \right\rangle$ 的平面的有效半径 r_{eff} 。
4. 给定一个含有 n 个顶点位置的数组, 编程构造它的边界盒、边界球、边界椭球以及边界圆柱。
5. 实现一个门系统, 可用任意凸多边形对视平截体进行裁剪, 并根据裁剪后的视平截体进行可视性测试。



第 8 章 碰撞检测



三维游戏要处理大量运动物体的各种动作。除非一个物体可以向周围环境发出一个力场,两个物体之间的交互作用通常发生在它们试图同时占据同一空间时。游戏引擎中,决定两个物体何时发生交互作用的过程称为碰撞检测。有两种典型的碰撞检测计算方法,其一是检测发生在一个运动物体与周围环境中静止的几何体之间的碰撞;其二是检测发生在两个都可能运动的物体之间的碰撞。本章将介绍几种确定碰撞发生时间与位置的技术。

8.1 环境碰撞

除了那些发生在无限空间中的游戏以外,大部分游戏需要确定运动物体和环境之间什么时候发生碰撞。为了减少碰撞检测的计算量,通常用边界体近似地表示具有复杂几何形状的运动物体。

假设在渲染某一帧场景的显示效果图时,已知某一运动物体的位置,同时也可以计算出在渲染下一帧效果图时,该物体将无障碍地运动到的位置。由于两帧之间的间隔通常非常短暂,可以将物体在两帧时间间隔之间视作为直线运动,因此,环境碰撞检测问题通常简化为边界体沿线段的延伸与环境中某一部分的求交问题。

小尺寸的运动物体通常被当作一个点来处理,这样就可以将碰撞检测问题简化成射线的求交问题。利用与可视性判断中类似的技术,大运动物体的碰撞检测计算也可以简化成射线的求交问题。例如,一个运动的球和一个平面之间的碰撞可以通过计算球心运动轨迹与一个平面(该平面为原平面沿其法线平移运动球半径的距离后形成的)之间的交点进行判断,可以由通过球心的路径和位移球半径距离的平面的相交检测到一个运动球体与一个平面的碰撞。

8.1.1 球体与平面的碰撞

给定一个半径为 r 的运动球体和任意一个在每一点都可以沿法线方向向外平移 r 的面,它们之间的碰撞检测问题可以用射线的求交问题替代。最简单的例子就是球体和平面之间的碰撞检测。如图 8.1 所示,当球体与平面 L 在正半空间相切时,球心 P 与平面间的距离为 r ,因此 $L \cdot P = r$ 。平面 L 的 4D 向量表示式为

$$L = \langle N, D \rangle \quad (8.1)$$

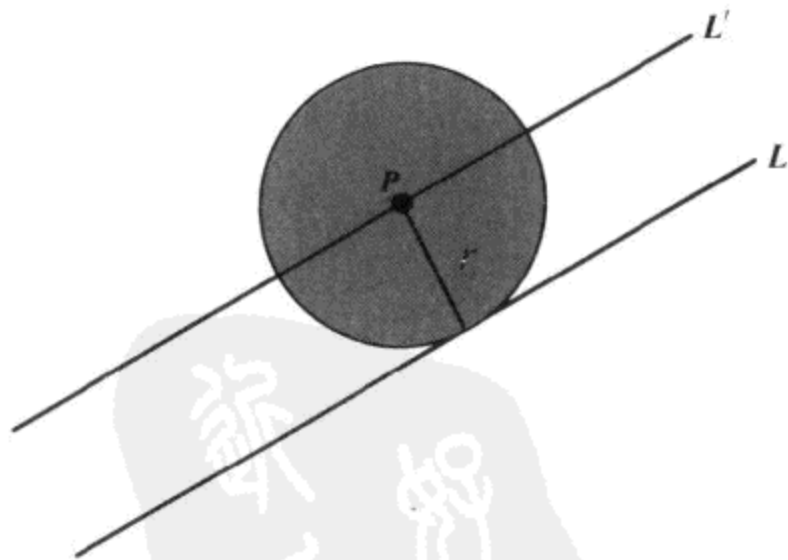


图 8.1 半径为 r 的球体与平面相切(其球心位于平面平移一定距离后形成的平面上)

关系式 $L \cdot P = r$ 可表示为

$$N \cdot P + D = r \quad (8.2)$$

将 r 移到等式左边,得

$$N \cdot P + D - r = 0 \quad (8.3)$$

即点 P 位于平面 L' 上

$$L' = \langle N, D - r \rangle \quad (8.4)$$

平面 L' 与 L 平行,它是平面 L 法线方向移动距离 r 后得到的。

给定半径为 r 的球,其球心从 $t=0$ 时的 P_1 点移动到 $t=1$ 时的 P_2 点,现要确定它是否与平面发生碰撞。假设球体最初与平面 L 不相交,起始点 P_1 位于平面正半空间(平面的负半空间代表某些对象内部),因此 $L \cdot P_1 \geq r$ 。若 $L \cdot P_1 \geq r$ 也成立,那么在时间区间 $0 \leq t \leq 1$ 内球仍在平面的正半空间,故没有发生碰撞。

球心在 t 时刻的位置 $P(t)$ 为

$$P(t) = P_1 + tV \quad (8.5)$$

其中, V 是球的速度

$$V = P_2 - P_1 \quad (8.6)$$

当式(8.7)成立,并且 t 的解满足 $0 \leq t \leq 1$ 时,球体与平面 $L = \langle N, D \rangle$ 发生碰撞

$$L' \cdot P(t) = 0 \quad (8.7)$$

其中, L' 为式(8.4)定义的平面。将等式(8.5)中的 $P(t)$ 代入得

$$L' \cdot P_1 + t(L' \cdot V) = 0 \quad (8.8)$$

解得

$$t = -\frac{L' \cdot P_1}{L' \cdot V} \quad (8.9)$$

因为向量 V 表示方向,其 w -坐标为 0,所以分母等于 $N \cdot V$ 。如果 $N \cdot V = 0$,那么球体与平面平行运动,不会发生相交,否则,球体与平面在 t 时间发生碰撞, t 值如式(8.9)所示。球与平面的切点 C 可由下式计算出来:

$$C = P(t) - rN \quad (8.10)$$

因此,该点位于与平面法线 N 方向相反的方向上并与球心距离为 r 的地方。

8.1.2 长方体与平面的碰撞

用与判断球体和平面碰撞类似的方法,可以确定一个运动的长方体是否与某个平面发生碰撞,不同的是这时平面平移的距离为长方体的有效半径(如 7.2.4 节中的介绍)。此外,长方体与平面可能在多点相切。不但长方体的整个边可能与平面发生碰撞,而且当长方体的整个面与平面平行接触时,也会发生碰撞。

假设一个长方体,其边的长度和方向为向量 R, S 和 T ,它对于法向量为 N 的平面的有效半径为 r_{eff}

$$r_{\text{eff}} = \frac{1}{2}(|R \cdot N| + |S \cdot N| + |T \cdot N|) \quad (8.11)$$

设 Q_1 为长方体中心在 $t=0$ 时刻的位置, Q_2 为长方体中心在 $t=1$ 时刻的位置,如图 8.2 所示,则长方体中心的位置函数为

$$Q(t) = Q_1 + tV \quad (8.12)$$

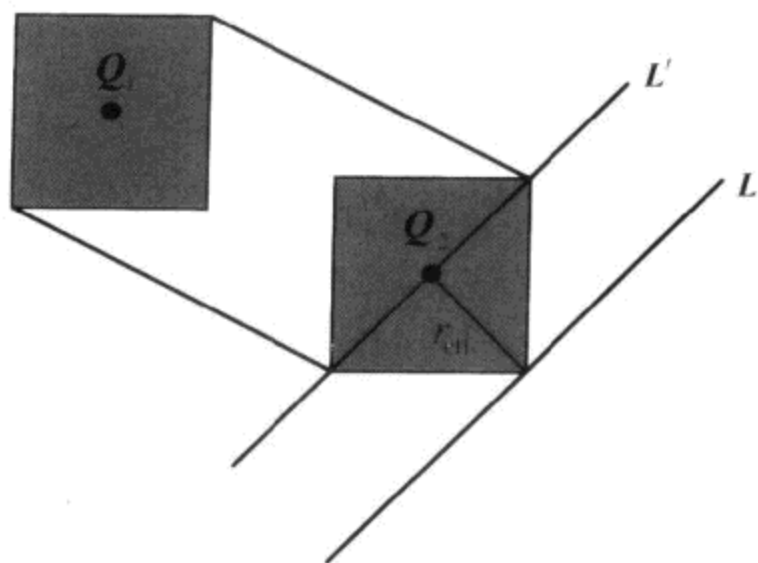


图 8.2 立方体盒与平面间的碰撞检查可以通过将平面平移其有效半径的距离来完成

其中 V 是长方体的运动速度

$$V = Q_2 - Q_1 \quad (8.13)$$

为了计算与平面 $L = \langle N, D \rangle$ 的交点, 需要计算出

$$t = -\frac{L' \cdot Q_1}{L' \cdot V} \quad (8.14)$$

其中平面 L' 是平行于平面 L , 并且沿平面 L 的法线方向平移距离 r_{eff} 后得到的

$$L' = \langle N, D - r_{\text{eff}} \rangle \quad (8.15)$$

假设长方体与平面在初始位置不相交, 长方体中心在 $t=0$ 时位于平面 L' 的正向空间 (也就是 $L' \cdot Q_1 \geq 0$)。因此, 如果 $L' \cdot Q_2 \geq 0$ 同样成立, 那么长方体仍然位于平面 L 的正半空间, 二者没有发生碰撞。

如果式(8.14)给出的 t 值满足 $0 \leq t \leq 1$, 就可确定长方体与平面确实发生了碰撞, 必须进一步确定碰撞发生在哪一点或哪些点。如果 $|R \cdot N|$, $|S \cdot N|$ 和 $|T \cdot N|$ 的值都不为零, 那么长方体的所有边都不与平面 L 平行, 不会发生边碰撞。在这种情况下, 碰撞一定发生在长方体的某个顶点, 通过分析长方体八个顶点的表达式, 可获得一个通用公式, 用它可以计算出与平面发生碰撞的顶点。立方体盒的顶点位置表示为

$$Z = Q(t) \pm \frac{1}{2}R \pm \frac{1}{2}S \pm \frac{1}{2}T \quad (8.16)$$

为了找到距离平面最近的点, 需要选择使点积 $L \cdot Z$ 最小的符号。当 $\pm R \cdot N$, $\pm S \cdot N$, $\pm T \cdot N$ 都为负时, 点积 $L \cdot Z$ 最小。当 $\pm R \cdot N$, $\pm S \cdot N$ 和 $\pm T \cdot N$ 中存在任一正值, 在式(8.16)就取相应的负号, 则碰撞点 C 可由下式得出:

$$C = Q(t) - \frac{1}{2}[\text{sgn}(R \cdot N)R + \text{sgn}(S \cdot N)S + \text{sgn}(T \cdot N)T] \quad (8.17)$$

其中, 函数 $\text{sgn}(x)$ 为符号函数, 返回 x 的符号

$$\text{sgn}(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases} \quad (8.18)$$

当 $|R \cdot N|$ 和 $|S \cdot N|$ 和 $|T \cdot N|$ 中仅有一个为零时, 它所对应的轴线就平行于平面, 碰撞一

定发生在长方体的某条边上。通过式(8.17)可得该边的两个端点 C_1 和 C_2 , 也就是式(8.17)中含有 0 的一项取正负两种情况而得到两个解。例如, 如果 $|T \cdot N|$ 为 0, 则 C_1 和 C_2 为

$$C_{1,2} = Q(t) - \frac{1}{2} [\text{sgn}(R \cdot N)R + \text{sgn}(S \cdot N)S \pm T] \quad (8.19)$$

当 $|R \cdot N|$, $|S \cdot N|$ 和 $|T \cdot N|$ 中有两个值为零, 需要对式(8.17)再作进一步的修改。此时, 碰撞发生在长方体的面上, 该面的顶点可由修改后的式(8.17), 也就是式(8.17)中含有 0 的两项分别取正负两种情况而得到四个解。例如, 如果, $|S \cdot N| = 0$ 且 $|T \cdot N| = 0$, 那么与平面发生碰撞的长方体面的四个顶点为

$$C_{1,2,3,4} = Q(t) - \frac{1}{2} [\text{sgn}(R \cdot N)R \pm S \pm T] \quad (8.20)$$

8.1.3 空间分割

确定一个物体是否与一个平面发生碰撞是在空间分割环境中快速检测到碰撞的基本要素。由于在用八叉树和 BSP 树表示的空间中, 区域是被平面分割开的, 通常可以无须将运动物体与场景大部分区域中的实际几何模型进行碰撞检查, 就可以知道它不与这些区域发生碰撞。

假设一个物体在一帧内由 P_1 点运动到 P_2 点, 令 $L = \langle N, D \rangle$ 表示以某种方式分割场景区的一个平面, 并且假设该运动物体相对于平面 L 的有效半径为 r_{eff} , 则如果物体位置 P 满足下列条件:

$$L \cdot P \geq r_{\text{eff}} \quad (8.21)$$

那么运动物体完全位于平面 L 的正半空间。而如果位置 P 满足

$$L \cdot P \leq -r_{\text{eff}} \quad (8.22)$$

则运动物体位于平面 L 的负半空间。如果表示运动物体位置的两点 P_1 和 P_2 都位于平面 L 的正半空间, 那么该物体的任何部分都不会穿过平面 L 到其负半空间中。同理, 如果表示运动物体位置的两点 P_1 和 P_2 都位于平面的负半空间, 则该物体的任何部分都不会穿过平面 L 到其正半空间。在这种情况下, 可以不必检测运动物体是否与位于平面 L 另一侧的几何体发生碰撞。

8.2 物体碰撞

这一节将讨论两个运动物体间的碰撞检测技术。通常两个运动的物体可以看成是一个物体相对于另一个物体的相对运动, 其相对运动的速度为两物体的速度之差, 因此运动物体间的碰撞检测问题可以简化为一个运动物体和一个固定物体之间的碰撞检测问题。下面将介绍运动的球体或立方体与静止的球体或立方体之间的碰撞检测方法。

8.2.1 两个球体之间的碰撞

假设两个运动着的球体, 在时间段从 $t=0$ 到 $t=1$ 之间有恒定的线性速度, 并假设开始时球体既不相交, 也没有出现一个球体包含另一个球体的情况。设点 P_1 和 P_2 为第一个球

体的球心在运动开始和结束时的位置,点 Q_1 和 Q_2 为第二个球体的球心在运动开始和结束时的位置,如图 8.3 所示,速度向量 V_P 和 V_Q 可定义为

$$\begin{aligned} V_P &= P_2 - P_1 \\ V_Q &= Q_2 - Q_1 \end{aligned} \quad (8.23)$$

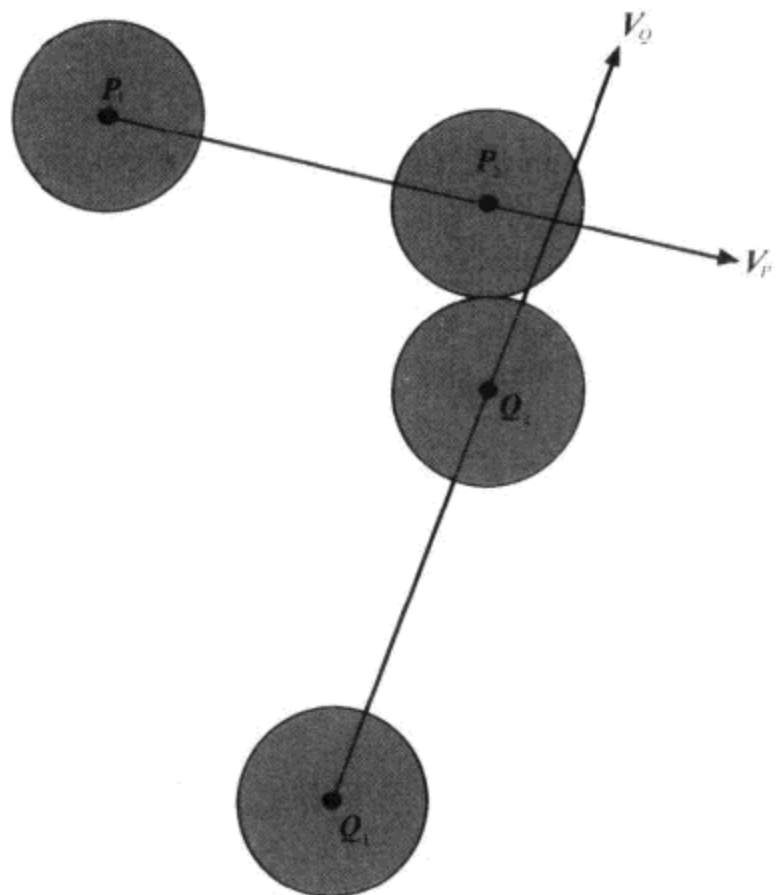


图 8.3 两个运动球体之间的碰撞检测

那么第一个球的球心位置 $P(t)$ 和第二个球的球心位置 $Q(t)$ 的表达式为

$$\begin{aligned} P(t) &= P_1 + tV_P \\ Q(t) &= Q_1 + tV_Q \end{aligned} \quad (8.24)$$

令 r_P 和 r_Q 分别为两个球的半径,需要判断球心 $P(t)$ 和 $Q(t)$ 之间的距离 d 是否在某一个 $t \in [0, 1]$ 的时刻等于 $r_P + r_Q$ 。如果成立,则在 t 时刻两球体相切,即发生了碰撞。实际的碰撞检测中,通常使用球心 $P(t)$ 和 $Q(t)$ 之间距离的平方值

$$d^2 = \|P(t) - Q(t)\|^2 \quad (8.25)$$

将式(8.24)中的 $P(t)$ 和 $Q(t)$ 带入得

$$d^2 = \|P_1 + tV_P - Q_1 - tV_Q\|^2 \quad (8.26)$$

为了便于表示,设

$$\begin{aligned} A &= P_1 - Q_1 \\ B &= V_P - V_Q \end{aligned} \quad (8.27)$$

则等式(8.26)可以写为

$$d^2 = \|A + tB\|^2 = A^2 + 2t(A \cdot B) + t^2B^2 \quad (8.28)$$

利用二次方程式解得 t 的值为

$$t_1 = \frac{-(A \cdot B) - \sqrt{(A \cdot B)^2 - B^2(A^2 - d^2)}}{B^2}$$

$$t_2 = \frac{-(\mathbf{A} \cdot \mathbf{B}) + \sqrt{(\mathbf{A} \cdot \mathbf{B})^2 - B^2(A^2 - d^2)}}{B^2} \quad (8.29)$$

如果两个球体相切,令 $d = r_P + r_Q$ 则可以解得两个球体发生碰撞的时间 t_1 和 t_2 。式(8.29)中根号下的表达式有可能为负值,此时两球体将不会发生碰撞。另外,也可能 $B^2 = 0$,则意味着两个球体均为静止或者沿着同一方向以相同的速率运动,故也不会发生碰撞。

由于 B^2 的值不为负, t_1 的值一定不会大于 t_2 的值。时间点 t_1 表示两球体在该时刻发生碰撞后仍然互相逐渐靠近,时间点 t_2 则表示在该时刻两球体发生碰撞后,就开始相互远离。由于,已经假设两球开始运动时不相交,因此,只须计算两球体第一次发生碰撞时的时间点 t_1 。也就是,只须用以下公式计算 t 即可检测出碰撞是否发生:

$$t = \frac{-(\mathbf{A} \cdot \mathbf{B}) - \sqrt{(\mathbf{A} \cdot \mathbf{B})^2 - B^2[A^2 - (r_P + r_Q)^2]}}{B^2} \quad (8.30)$$

如果 t 的值不属于区间 $[0, 1)$,那么在该时间间隔中两球体就没有发生碰撞。

不计算式(8.30)也可以检测出没有碰撞发生。将方程(8.28)对于 t 的偏导数设为 0,就可以求出使距离平方 d^2 最小时的时间点 t ,即

$$2B^2t + 2(\mathbf{A} \cdot \mathbf{B}) = 0 \quad (8.31)$$

解上述方程当两球心距离最小时 t 的值为

$$t = -\frac{\mathbf{A} \cdot \mathbf{B}}{B^2} \quad (8.32)$$

带入式(8.28)得到两个球心之间的最短距离为

$$d^2 = A^2 - \frac{(\mathbf{A} \cdot \mathbf{B})^2}{B^2} \quad (8.33)$$

如果 $d^2 > (r_P + r_Q)^2$,则两个球体不会发生碰撞。

当检测到在 t 时刻两球体发生了碰撞,就可以将这个 t 值代入等式(8.24)计算两球心的位置 $\mathbf{P}(t)$ 和 $\mathbf{Q}(t)$ 。如图 8.4 所示,切点 C 位于 $\mathbf{P}(t)$ 和 $\mathbf{Q}(t)$ 之间的线段上距离 $\mathbf{P}(t)$ 为 r_P 的地方,可以表示为

$$\mathbf{C} = \mathbf{P}(t) + r_P \mathbf{N} \quad (8.34)$$

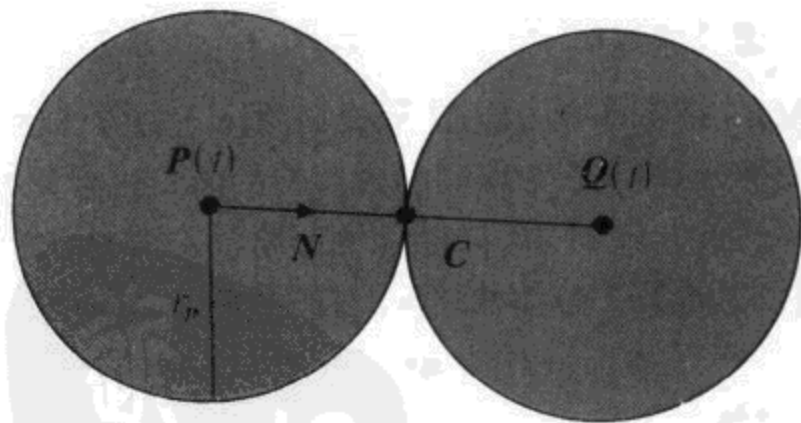


图 8.4 两个球体的切点 C 位于发生碰撞时两球心间的连线上

其中 N 为由 $\mathbf{P}(t)$ 指向 $\mathbf{Q}(t)$ 的单位法向量。

$$\mathbf{N} = \frac{\mathbf{Q}(t) - \mathbf{P}(t)}{\|\mathbf{Q}(t) - \mathbf{P}(t)\|} \quad (8.35)$$

8.2.2 球体与长方体的碰撞

假设一个半径为 r 的球体,其球心以恒定速率由时间 $t=0$ 时刻的点 P_1 运动到时间 $t=1$ 时刻的点 P_2 ,一个边的尺寸和方向分别为向量 R, S 和 T 的长方体,其中心由时间 $t=0$ 时刻的 Q_1 点运动到 $t=1$ 时刻的 Q_2 点,如图 8.5 所示。同样,假设开始运动时球体与长方体互不相交,也没有出现相互包含的情况。

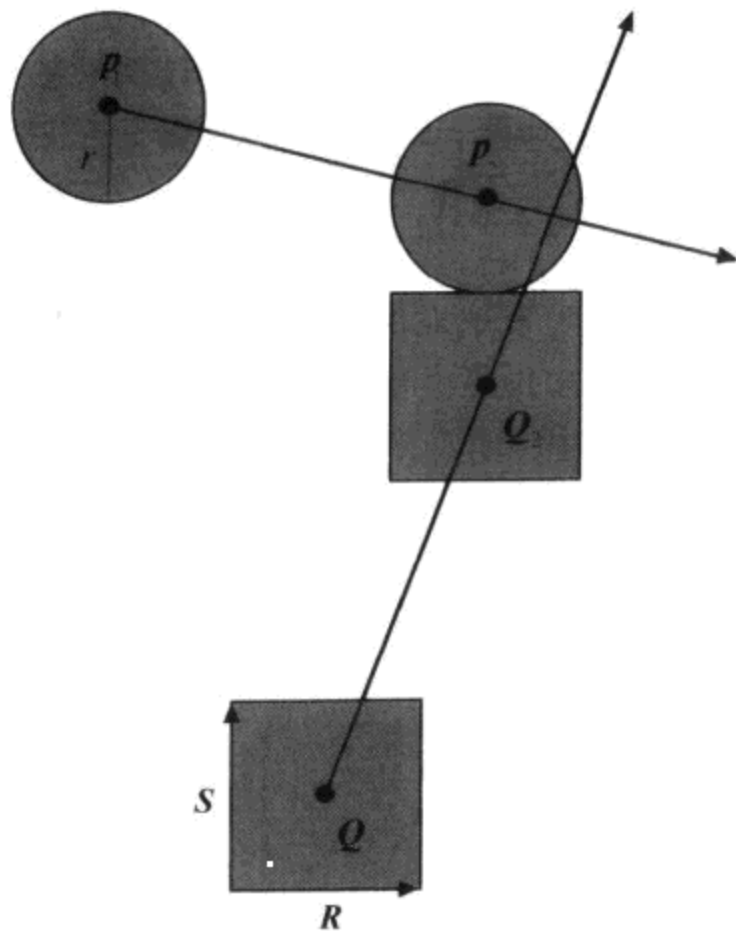


图 8.5 一个运动的球体和一个运动的长方体之间的碰撞检测

由于确定球体与长方体之间没有发生碰撞的必要的计算量较大,通常首先用 8.2.1 一节中介绍的方法检测长方体的外接球体是否与该球体发生了碰撞(长方体的外接球体的半径为 $\frac{1}{2} \|R + S + T\|$)。如果球体从 P_1 点运动到 P_2 点,长方体从 Q_1 点运动到 Q_2 点的过程中,球体未与长方体的外接球体发生碰撞,那么该球体与长方体也不可能发生碰撞。

通过相应的坐标变换,球体与长方体间的碰撞检测问题可以简化为判断运动球体与一个静止的与坐标轴对齐的长方体之间的碰撞检测问题。首先,从球体的运动速度中减去长方体的运动速度 V ,得到基准帧中球体的速度,在该帧中长方体是静止的

$$V = (P_2 - P_1) - (Q_2 - Q_1) \quad (8.36)$$

基准帧中球心位置 $P(t)$ 为

$$P(t) = P_1 + tV \quad (8.37)$$

而长方体的中心始终位于 Q_1 。

现在要通过坐标变换将静止的长方体与坐标轴对齐。由于向量 R, S 和 T 是相互垂直的,以下列单位向量为行向量可以构成一个 3×3 矩阵

$$\hat{R} = \frac{\mathbf{R}}{\|\mathbf{R}\|}, \quad \hat{S} = \frac{\mathbf{S}}{\|\mathbf{S}\|}, \quad \hat{T} = \frac{\mathbf{T}}{\|\mathbf{T}\|} \quad (8.38)$$

该矩阵可将长方体的边(用向量 \mathbf{R}, \mathbf{S} 和 \mathbf{T} 表示)映射到 x, y 和 z 轴上。另外,还要将下述顶点 G 平移到坐标系原点:

$$\mathbf{G} = Q_1 - \frac{1}{2}(\mathbf{R} + \mathbf{S} + \mathbf{T}) \quad (8.39)$$

这样,可以得到最后的变换矩阵 \mathbf{M}

$$\mathbf{M} = \begin{bmatrix} \hat{R}_x & \hat{R}_y & \hat{R}_z & -\hat{\mathbf{R}} \cdot \mathbf{G} \\ \hat{S}_x & \hat{S}_y & \hat{S}_z & -\hat{\mathbf{S}} \cdot \mathbf{G} \\ \hat{T}_x & \hat{T}_y & \hat{T}_z & -\hat{\mathbf{T}} \cdot \mathbf{G} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8.40)$$

用矩阵 \mathbf{M} 对长方体进行变换后,组成长方体的 6 个面分别为

$$\begin{aligned} x = 0 & \quad x = \|\mathbf{R}\| \\ y = 0 & \quad y = \|\mathbf{S}\| \\ z = 0 & \quad z = \|\mathbf{T}\| \end{aligned} \quad (8.41)$$

球心的起始位置 \mathbf{P}_1 变换到新坐标系中的位置 \mathbf{P}'_1 为

$$\mathbf{P}'_1 = \mathbf{M}\mathbf{P}_1 \quad (8.42)$$

在新坐标系中球体的速度 \mathbf{V}' 为

$$\mathbf{V}' = \mathbf{M}_{3 \times 3} \mathbf{V} \quad (8.43)$$

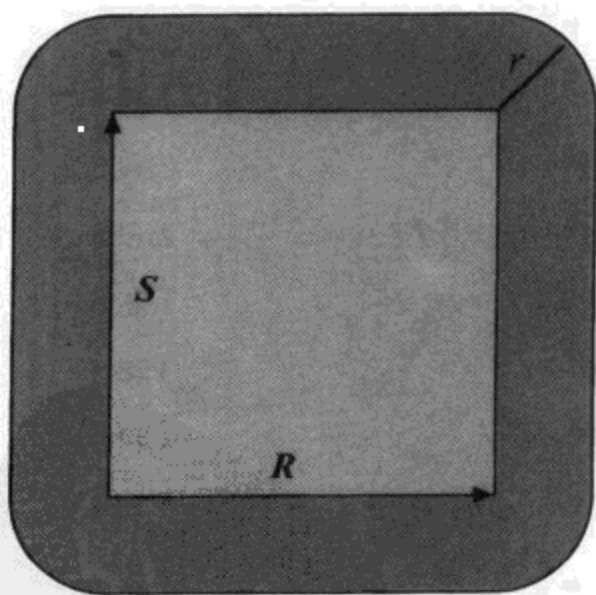
由于向量 \mathbf{V} 表示速度的方向,它的 4D 坐标表示中含有一个为 0 的 w -坐标,因此,速度向量 \mathbf{V} 可以仅用变换矩阵 \mathbf{M} 中进行旋转变换的部分进行变换。

现在,接下来的任务是检测一个运动的球体与一个静止的与坐标轴对齐的长方体之间是否发生碰撞,在时刻 t ,运动球体的球心位置为

$$\mathbf{P}'(t) = \mathbf{P}'_1 + t\mathbf{V}' \quad (8.44)$$

球体与正方体盒的碰撞可能发生在长方体的一个面上、边上或者顶点处。当球体与长方体碰撞时,其中心位于与长方体表面某个点相距为 r 的位置。因此,如果球体与长方体发生碰撞,就需要计算出该时刻当球心位于与长方体表面距离为 r 的点集合中。如图 8.6 所示,这个点集合是由 6 个平面、12 个与长方体的棱边相对应的圆柱面以及 8 个与长方体顶点相对应的球面组成。由于长方体是与坐标轴对齐,这 26 个面可以用简单的数学表达式表示。

6 个与长方体表面的距离为 r 的平面可以用平面方程和表 8.1 中所列出的边界条件表示。因为与速度向量 \mathbf{V}' 的



注:为了检测一个球体是否与一个长方体发生了碰撞,根据球半径将长方体扩大,并且计算球心直线轨迹与扩展后的长方体表面相交情况

图 8.6 检测一个球体是否与一个长方体发生了碰撞

点积为非负值的平面的法线远离球体,所以它不会包含与球体首先发生碰撞的点,因此我们最多只考虑与运动球体的球心轨迹可能相交的 3 个平面。

表 8.2 中列出了 12 个以长方体边为轴线并且半径为 r 的 $1/4$ 圆柱,它们的轴线分别与 x, y 和 z 坐标轴平行。 $1/4$ 圆柱面被边界条件限制在两个邻接平面之间的部分。与不参加碰撞检测的 3 个面对应,对于那些轴线为两不可见平面交线的 $1/4$ 圆柱面,由于面上的任意点都位于远离运动球体的运动轨迹的方向上,因此这些面也不参与运动球体的碰撞检测。由于一定存在三个两两相邻的平面不参加碰撞检测,故至少有三个 $1/4$ 圆柱面也不参加碰撞检测。因此,最多有 9 个 $1/4$ 圆柱面参加与球体的运动轨迹之间的求交运算。

表 8.1 将长方体扩展球体半径 r 后长方体的 6 个面的表达式

面	平面	法线	边界条件	
1	$x = -r$	$\langle -1, 0, 0 \rangle$	$0 \leq y \leq \ S\ $	$0 \leq z \leq \ T\ $
2	$x = \ R\ + r$	$\langle 1, 0, 0 \rangle$	$0 \leq y \leq \ S\ $	$0 \leq z \leq \ T\ $
3	$y = -r$	$\langle 0, -1, 0 \rangle$	$0 \leq x \leq \ R\ $	$0 \leq z \leq \ T\ $
4	$y = \ S\ + r$	$\langle 0, 1, 0 \rangle$	$0 \leq x \leq \ R\ $	$0 \leq z \leq \ T\ $
5	$z = -r$	$\langle 0, 0, -1 \rangle$	$0 \leq x \leq \ R\ $	$0 \leq y \leq \ S\ $
6	$z = \ T\ + r$	$\langle 0, 0, 1 \rangle$	$0 \leq x \leq \ R\ $	$0 \leq y \leq \ S\ $

表 8.2 扩展半径 r 后的长方体中的 12 个圆柱面

圆柱面	公共面	边界条件		
$x^2 + y^2 = r^2$	1 和 3	$x < 0$	$y < 0$	$0 \leq z \leq \ T\ $
$(x - \ R\)^2 + y^2 = r^2$	2 和 3	$x > \ R\ $	$y < 0$	$0 \leq z \leq \ T\ $
$(x - \ R\)^2 + (y - \ S\)^2 = r^2$	2 和 4	$x > \ R\ $	$y \geq \ S\ $	$0 \leq z \leq \ T\ $
$x^2 + (y - \ S\)^2 = r^2$	1 和 4	$x < 0$	$y \geq \ S\ $	$0 \leq z \leq \ T\ $
$x^2 + z^2 = r^2$	1 和 5	$x < 0$	$z < 0$	$0 \leq y \leq \ S\ $
$(x - \ R\)^2 + z^2 = r^2$	2 和 5	$x > \ R\ $	$z < 0$	$0 \leq y \leq \ S\ $
$(x - \ R\)^2 + (z - \ T\)^2 = r^2$	2 和 6	$x > \ R\ $	$z > \ T\ $	$0 \leq y \leq \ S\ $
$x^2 + (z - \ T\)^2 = r^2$	1 和 6	$x < 0$	$z > \ T\ $	$0 \leq y \leq \ S\ $
$y^2 + z^2 = r^2$	3 和 5	$y < 0$	$z < 0$	$0 \leq x \leq \ R\ $
$(y - \ S\)^2 + z^2 = r^2$	4 和 5	$y \geq \ S\ $	$z < 0$	$0 \leq x \leq \ R\ $
$(y - \ S\)^2 + (z - \ T\)^2 = r^2$	4 和 6	$y \geq \ S\ $	$z \geq \ T\ $	$0 \leq x \leq \ R\ $
$y^2 + (z - \ T\)^2 = r^2$	3 和 6	$y < 0$	$z \geq \ T\ $	$0 \leq x \leq \ R\ $

注:公共面表示表 8.1 中所列的面中哪两个面与圆柱面共边

表 8.3 中列出了半径为 r 且球心位于长方体顶点的 8 个球面。在边界条件的限制下,每个球面实际只为 $1/8$ 球面。与 3 个不参加碰撞检测的面对应,三个面的公共顶点处的 $1/8$ 球面,由于面上的任意点都位于远离运动球体的运动轨迹的方向上,而不参加与运动球体的

碰撞检测。至少有一个顶点对应的 1/8 球面要被排除在碰撞检测之外,这样最多需要检测 7 个球面是否与球体运动轨迹相交。

表 8.3 扩展半径 r 后的长方体中的 8 个 1/8 球面

球面	公共面	边界条件		
$x^2 + y^2 + z^2 = r^2$	1,3 和 5	$x < 0$	$y < 0$	$z < 0$
$(x - \ R\)^2 + y^2 + z^2 = r^2$	2,3 和 5	$x > \ R\ $	$y < 0$	$z < 0$
$(x - \ R\)^2 + (y - \ S\)^2 + z^2 = r^2$	2,4 和 5	$x > \ R\ $	$y \geq \ S\ $	$z < 0$
$x^2 + (y - \ S\)^2 + z^2 = r^2$	1,4 和 5	$x < 0$	$y \geq \ S\ $	$z < 0$
$x^2 + y^2 + (z - \ T\)^2 = r^2$	1,3 和 6	$x < 0$	$y < 0$	$z > \ T\ $
$(x - \ R\)^2 + y^2 + (z - \ T\)^2 = r^2$	2,3 和 6	$x > \ R\ $	$y < 0$	$z > \ T\ $
$(x - \ R\)^2 + (y - \ S\)^2 + (z - \ T\)^2 = r^2$	2,4 和 6	$x > \ R\ $	$y > \ S\ $	$z > \ T\ $
$x^2 + (y - \ S\)^2 + (z - \ T\)^2 = r^2$	1,4 和 6	$x < 0$	$y > \ S\ $	$z > \ T\ $

注:公共面表示表 8.1 中所列的面中公共顶点为球面球心的三个面

通过判断运动球体的球心运动轨迹 $P(t)$ 与扩展后的长方体是否相交,可以完成球体与长方体之间的碰撞检测。首先,我们检测球心轨迹 $P(t)$ 与表 8.1 中所示的平面中的至多三个符合条件的平面是否相交(任何法线与速度向量 V' 的点积为非数值的平面都要被排除在碰撞检测之外)。如果没有交点,那么接着检测表 8.2 中所示的 1/4 圆柱面中符合条件的至多 9 个圆柱体。如果仍然没有交点,则最后检测表 8.3 中所示 1/8 球面中符合条件的最多 7 个球体。最终,如果没有交点则证明运动球体与长方体没有发生碰撞。

球心轨迹与表 8.1 中所示的平面之间的交点计算方法如下,通过将球心轨迹 $P(t)$ 的分量 x, y 和 z 中的相应分量带入特定的平面方程,可以求出相应的时间。下面的例子就说明了计算球心轨迹 $P(t)$ 与表 8.1 中所示的第一个平面的交点的计算方法。

【例 8.1】 确定球心轨迹 $P'(t) = P'_1 + V'$ 与平面 $x = -r$ 是否相交,其中平面的边界条件为 $0 \leq y \leq \|S\|$ 和 $0 \leq z \leq \|T\|$ 。

解 将球心轨迹 $P'(t)$ 的 x -分量作为 x 变量带入平面方程中得

$$P'_1 + tV'_x = -r \tag{8.45}$$

解 t , 可得可能发生碰撞的时刻 t_0 的表达式为

$$t_0 = -\frac{r + (P'_1)_x}{V'_x} \tag{8.46}$$

其中,如果 V'_x 等于 0,则在碰撞检测中就可以不考虑这个平面,这是因为平面法线 $\langle -1, 0, 0 \rangle$ 与 V' 的点积为 0。如果 t_0 不属于区间 $[0, 1)$,则在此时刻没有发生碰撞,否则,必须检查轨迹 $P'(t_0)$ 的 y 分量和 z 分量是否满足不等式

$$\begin{aligned} 0 &\leq (P'_1)_y + t_0 V'_y \leq \|S\| \\ 0 &\leq (P'_1)_z + t_0 V'_z \leq \|T\| \end{aligned} \tag{8.47}$$

上述两个不等式都成立时,球心轨迹 $P'(t)$ 与长方体的这个面才相交。

若已经检测到了球心轨迹与扩展长方体中的平面相交,那就无需再作进一步的计算就

可以确定产生了碰撞。而如果球心轨迹与 6 个平面都没有相交,就需要继续检测运动球体是否与长方体的边发生碰撞,也就是判断球心轨迹 $P'(t)$ 上相应线段是否与表 8.2 中给出的圆柱面相交。与检测平面的方法类似,将 $P'(t)$ 的 x, y 和 z 分量代入特定的圆柱体方程以求出时间 t 。下面例子说明了检测运动球体与表 8.2 中所列的第一个圆柱面的交点的计算方法。

【例 8.2】 判断球心轨迹 $P'(t) = P'_1 + tV'$ 与圆柱面 $x^2 + y^2 = r^2$ 的相交情况,其中圆柱面的边界条件为 $x < 0, y < 0$ 和 $0 \leq z \leq \|T\|$ 。

解 将球心轨迹 $P'(t)$ 的 x 分量与 y 分量分别作为变量 x 和 y 代入圆柱方程中得

$$(P'_1 + tV')_x^2 + (P'_1 + tV')_y^2 = r^2 \quad (8.48)$$

展开可得到以下关于 t 的二次方程

$$(V'_x{}^2 + V'_y{}^2)t^2 + 2[(P'_1)_x V'_x + (P'_1)_y V'_y]t + (P'_1)_x^2 + (P'_1)_y^2 - r^2 = 0 \quad (8.49)$$

通过判别式 $D = b^2 - 4ac$, 可以很快地判断出线段与圆柱面是否相交,其中

$$\begin{aligned} a &= V'_x{}^2 + V'_y{}^2 \\ b &= 2[(P'_1)_x V'_x + (P'_1)_y V'_y] \\ c &= (P'_1)_x^2 + (P'_1)_y^2 - r^2 \end{aligned} \quad (8.50)$$

如果 $D \leq 0$, 则线段与圆柱面不相交。如果 $D = 0$, 球体刚好擦过长方体的边。当 $D > 0$ 时, 由于 a 为正值, 则取较小的时间 t_0 为碰撞发生的时刻

$$t_0 = \frac{-b - \sqrt{D}}{2a} \quad (8.51)$$

t_0 的较大解对我们来说没有意义, 这是因为它表示从起始点 P'_1 观察圆柱面时, 球心轨迹与该 1/4 圆柱面关于轴线对称的另一 1/4 柱面的交点。如果由 8.51 式求出的 t_0 值不属于 $[0, 1)$ 区间, 则在该时间区间内没有发生碰撞。如果值属于 $[0, 1)$ 区间, 那么需要判断以下不等式是否成立以最终确定 $P'(t_0)$ 点是否位于该 1/4 圆柱面上

$$0 \leq (P'_1)_z + t_0 V'_z \leq \|T\| \quad (8.52)$$

由于扩展后的长方体是一个凸多面体, 所以不必校验另外两个边界条件 $x < 0$ 和 $y < 0$ 是否成立。这一步中计算出的线段 $P'(t)$ 与圆柱面的任一交点都不会位于长方体中某个表面的后面。

如果球心轨迹 $P'(t)$ 与某个圆柱面有交点, 就无需再计算下去了, 否则, 就要开始计算球心轨迹 $P'(t)$ 与表 8.3 中所列出来的球面是否存在交点。同样, 将 $P'(t)$ 的 x, y 和 z 分量代入待测试的球面方程以求出它们的交点。下面的例子说明了检测运动球体与表 8.3 中所列的第一个球面交点的计算方法。

【例 8.3】 判断球心轨迹 $P'(t) = P'_1 + tV'$ 与球面 $x^2 + y^2 + z^2 = r^2$ 是否相交, 其中球面的边界条件为 $x < 0, y < 0$ 和 $z < 0$ 。

解 将球心轨迹 $P'(t)$ 的 x, y 与 z 分量分别作为变量 x, y 和 z 代入球面方程得

$$(P'_1 + tV')_x^2 + (P'_1 + tV')_y^2 + (P'_1 + tV')_z^2 = r^2 \quad (8.53)$$

如 5.2.3 一节中所述, 射线与球面的交点可以通过解二次方程 $at^2 + bt + c = 0$ 得到, 其中

$$\begin{aligned} a &= V'^2 \\ b &= 2(P'_1 \cdot V') \end{aligned}$$

$$c = P_1'^2 - r^2 \quad (8.54)$$

计算判别式 $D = b^2 - 4ac$ 的值可知是否发生碰撞,如果 $D > 0$,则较小的解 t_0 为

$$t_0 = \frac{-b - \sqrt{D}}{2a} \quad (8.55)$$

如果 $0 \leq t_0 \leq 1$,则发生碰撞,不必再检测点 $P'(t_0)$ 是否满足球面的边界条件,因为任何与球面的交点位于扩展长方体内部的直线段一定先与该长方体一个平面或圆柱面相交了。因此,点 $P'(t_0)$ 一定位于这个组成扩展长方体表面的 $1/8$ 球面之上。

如果球心轨迹 $P'(t)$ 与扩展长方体在 t_0 时刻相交,那么该时刻球心和长方体中心原始坐标系中的位置可以简单地通过计算 t_0 时刻的 $P(t)$ 和 $Q(t)$ 得到。要计算出 C 切点的实际位置,首先需要确定切点的法向量。如果球体与长方体的平面发生碰撞,则变换后的坐标系中平面的法向量 N' 为表 8.1 中列出的平面法向量。如果运动球体与表示长方体的棱边的 $1/4$ 圆柱面或表示长方体顶点的 $1/8$ 球面发生碰撞,则切点的法向量 N' 可以通过计算相应的平面方程在点 $P'(t_0)$ 处的一阶导数得出。由于矩阵 M 中表示旋转的 3×3 的矩阵是正交的,则原坐标系中切点的法向量 N 为

$$N = M^T N' \quad (8.56)$$

切点 C 即为

$$C = P(t_0) - rN \quad (8.57)$$

8.2.3 长方体之间的碰撞

假设两个任意方向的长方体 P 和 Q ,它们以恒定线性速率运动,如图 8.7 所示。设点 P_1 和 P_2 为第一个长方体的中心在运动开始($t=0$)和结束($t=1$)时的位置,点 Q_1 和 Q_2 为第二个长方体的中心在运动开始($t=0$)和结束($t=1$)时的位置。令第一个长方体的 3 条棱边大小和方向为向量 R_P, S_P 和 T_P ,第二个长方体的 3 条棱边大小和方向为向量 R_Q, S_Q 和 T_Q 。另外,仍然假设初始时两个长方体互不相交,也互不包含。

由于直接计算两个运动的长方体之间碰撞情况比较复杂,通常首先检测两个长方体的外接球体之间的碰撞情况。如果这两个球体并未发生碰撞,就不需要再进行更为复杂的长方体的碰撞检测计算。

两个长方体之间发生碰撞的方式有多种,可以将其归类为以下四种:

- (1) 一个长方体的顶点与另一个长方体的面发生碰撞。包括一个长方体的顶点与另一个长方体的棱边或者顶点发生碰撞的情况。

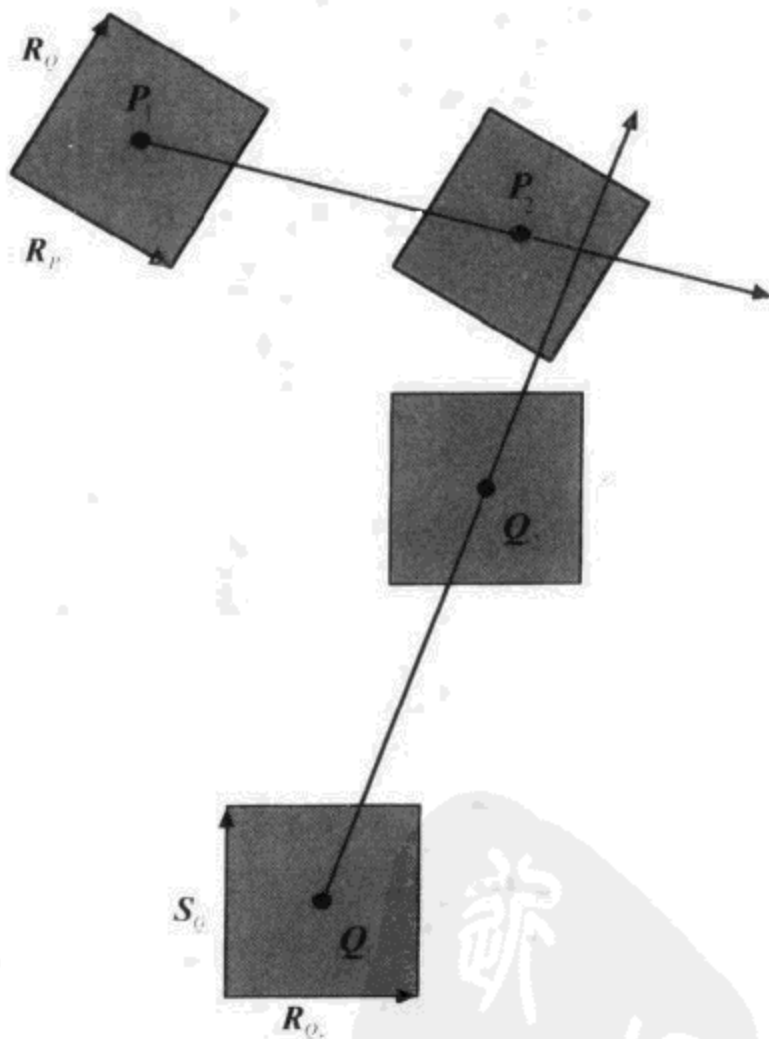


图 8.7 两个长方体之间的碰撞检测

- (2) 一个长方体的棱边与另一个长方体的面发生碰撞。包括一个长方体的棱边与另一个长方体的棱边发生碰撞并且两条棱边平行的情况。
- (3) 一个长方体的面与另一个长方体的面发生碰撞,且两个面平行。
- (4) 一个长方体的棱边与另一个长方体的棱边在某一个点上发生碰撞,并且发生碰撞的两个棱边都不与另一个长方体的任何一个面平行。

前三种可能情况都可以应用 8.1.2 一节中介绍的长方体-平面碰撞检测的方法来检测一个长方体与另一个长方体的面是否发生碰撞。这些检测是相互的,也就是,既需要检测长方体 P 相对与长方体 Q 的平面是否发生碰撞,又需要检测长方体 Q 相对与长方体 P 的平面是否发生碰撞。听起来计算的任务很繁重,实际上至多只有一半的长方体的面可能发生这类碰撞。在这一节中,首先介绍长方体-平面之间碰撞检测。大多数情况下,在这一步骤中就可以明确得出两个长方体之间发生碰撞或者完全不会发生碰撞的结论。然而,在某些特殊情况中,需要继续执行检测上述的第(4)类碰撞的算法,即检测一个长方体的棱边与另一个长方体的棱边在某一个点上发生的碰撞,同时发生碰撞的两个棱边都不与另一个长方体的任何一个面平行。

碰撞检测方法要在两个坐标系中分别进行计算。在第一个坐标系中,长方体 P 经过坐标变换后为一与坐标轴对齐的静止长方体,而在第二个坐标系中,长方体 Q 经过坐标变换后为一与坐标轴对齐的静止长方体。与球体-长方体间碰撞检测的方法相同,可以通过将长方体 Q 的速度 V 和两个长方体进行以下变换得到一个新的坐标空间,该空间中长方体 P 静止并且与坐标轴对齐。

$$V = (Q_1 - Q_1) - (P_2 - P_1) \quad (8.58)$$

两个长方体 P 和 Q 的变换矩阵 M 为

$$M = \begin{bmatrix} (\hat{R}_P)_x & (\hat{R}_P)_y & (\hat{R}_P)_z & -\hat{R}_P \cdot G_P \\ (\hat{S}_P)_x & (\hat{S}_P)_y & (\hat{S}_P)_z & -\hat{S}_P \cdot G_P \\ (\hat{T}_P)_x & (\hat{T}_P)_y & (\hat{T}_P)_z & -\hat{T}_P \cdot G_P \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8.59)$$

其中 \hat{R}_P, \hat{S}_P 和 \hat{T}_P 是单位向量,定义如下

$$\hat{R}_P = \frac{\mathbf{R}_P}{\|\mathbf{R}_P\|}, \quad \hat{S}_P = \frac{\mathbf{S}_P}{\|\mathbf{S}_P\|}, \quad \hat{T}_P = \frac{\mathbf{T}_P}{\|\mathbf{T}_P\|} \quad (8.60)$$

其中点 G_P 为

$$G_P = P_1 - \frac{1}{2}(\mathbf{R}_P + \mathbf{S}_P + \mathbf{T}_P) \quad (8.61)$$

则新坐标空间中长方体 P 的 6 个面分别为

$$\begin{aligned} \langle -1, 0, 0, 0 \rangle & \quad \langle 1, 0, 0, -\|\mathbf{R}_P\| \rangle \\ \langle 0, -1, 0, 0 \rangle & \quad \langle 0, 1, 0, -\|\mathbf{S}_P\| \rangle \\ \langle 0, 0, -1, 0 \rangle & \quad \langle 0, 0, 1, -\|\mathbf{T}_P\| \rangle \end{aligned} \quad (8.62)$$

这 6 个平面的方程与方程 8.41 给出的平面的方程等效。在新坐标空间中长方体 Q 的中心的起始位置 Q'_1 为

$$Q'_1 = MQ_1 \quad (8.63)$$

该长方体的棱边向量 R'_Q, S'_Q 和 T'_Q 分别为

$$\begin{aligned} R'_Q &= M_{3 \times 3} R_Q \\ S'_Q &= M_{3 \times 3} S_Q \\ T'_Q &= M_{3 \times 3} T_Q \end{aligned} \quad (8.64)$$

长方体 Q 的中心位置 $Q(t)$ 为

$$Q(t) = Q'_1 + tV' \quad (8.65)$$

其中 V' 是变换后的速度

$$V' = M_{3 \times 3} V \quad (8.66)$$

现在,要确定长方体 P 的哪些平面要参加碰撞检测的计算。首先可以排除那些法向量与速度 V' 的点积不为正的平面,因为这些平面远离另一长方体 Q 的运动方向。这样在长方体 P 的 6 个面中至多有 3 个面参与碰撞检测计算。另外,对于长方体 P 一个面所在的平面 $L = \langle N, D \rangle$,如果长方体 Q 一开始就完全位于该平面的负半空间,则长方体中的相应面也被排除在碰撞检测之外。首先计算长方体 Q 对于平面 L 的有效半径 r_{eff} ,由于式(8.62)所示的平面方程都很简单,有效半径 r_{eff} 的计算量很少。如果 $L \cdot Q_1 \leq r_{\text{eff}}$,则长方体 Q 位于平面 L 的负半空间。如果经过以上计算,长方体 P 的 6 个面都不会与长方体 Q 发生碰撞,则两个运动的长方体之间的运动为相互远离运动,因此不会发生碰撞。

对于长方体 P 中符合碰撞检查条件的面 F ,可以对长方体 Q 和包含这个面的平面 L 执行长方体-平面间的碰撞检测。如果长方体 Q 开始就完全位于平面 L 的正半空间,即 $L \cdot Q_1 \geq r_{\text{eff}}$,且与整个平面 L 没有发生碰撞,那么长方体 P 与长方体 Q 也不会发生碰撞,则可以立即结束长方体-长方体之间的碰撞检测(如果长方体 Q 开始就与平面 L 有交点,那此时还不能停止碰撞检测的计算)。否则,继续判断长方体 Q 和平面 L 之间发生的碰撞是否位于面 F 之内。假设碰撞发生在一个点 C ,则需要判断该点是否落入与面 F 垂直的两对相对的面之内。例如,通过碰撞检测发现长方体 Q 与平面 $\langle -1, 0, 0, 0 \rangle$ 在 C 点发生碰撞,必须校验以下条件是否成立:

$$\begin{aligned} 0 &\leq C_y \leq \|S_P\| \\ 0 &\leq C_z \leq \|T_P\| \end{aligned} \quad (8.67)$$

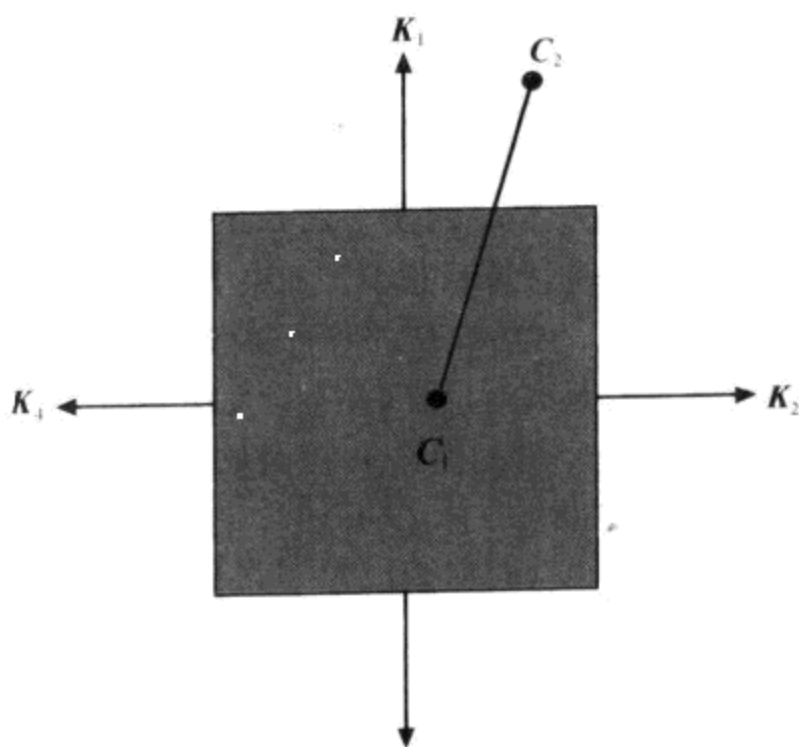
如果上式不成立,则证明碰撞并没有发生在长方体面 F 之内。如果碰撞的确发生在面 F 之内,则碰撞检测就可以停止,这是因为不会在另一个面上发生更早的碰撞。

如果长方体 Q 与平面 L 在端点为 C_1 和 C_2 的边上发生碰撞,则必须确定该线段的某一部分是否位于面 F 之内,通过用与面 F 垂直的四个面对线段进行裁剪可以完成这个判断,如图 8.8 所示。对每一个与平面 L 垂直的平面 K ,计算 $K \cdot C_1$ 和 $K \cdot C_2$,如果这两个值都为正,那么这个线段落在面 F 之外,并且没有发生碰撞。如果两个值都为负,则开始计算下一个平面。如果一个值为正一个值为负,则用线段与此平面的交点代替位于平面 K 正半空间的端点,并且开始计算下一个平面。如果对每一个平面都进行计算之后,线段还没有被裁剪完,则一定发生了碰撞,碰撞检测停止。

最后一种情况是长方体 Q 与平面 L 在由顶点 C_1, C_2, C_3 和 C_4 构成的平面上发生碰撞,对于这种情况必须确定由这几个顶点组成的四边形是否有完全或者部分位于面 F 之内。通过运用 9.2.2 一节中介绍的方法,用与面垂直的四个面裁剪这个四边形可以完成上

述判断。为了保证算法的正确性,顶点 C_1 、 C_2 、 C_3 和 C_4 在顶点列表中的顺序必须按邻接关系连续排放。通过选择方程 8.20 中的适当符号可以保证顶点的排列顺序。

$$\begin{aligned} C_1 &= Q'(t) - \frac{1}{2} [\operatorname{sgn}(\mathbf{R}_Q' \cdot \mathbf{N}) \mathbf{R}_Q' + \mathbf{S}_Q' + \mathbf{T}_Q'] \\ C_2 &= Q'(t) - \frac{1}{2} [\operatorname{sgn}(\mathbf{R}_Q' \cdot \mathbf{N}) \mathbf{R}_Q' - \mathbf{S}_Q' + \mathbf{T}_Q'] \\ C_3 &= Q'(t) - \frac{1}{2} [\operatorname{sgn}(\mathbf{R}_Q' \cdot \mathbf{N}) \mathbf{R}_Q' - \mathbf{S}_Q' - \mathbf{T}_Q'] \\ C_4 &= Q'(t) - \frac{1}{2} [\operatorname{sgn}(\mathbf{R}_Q' \cdot \mathbf{N}) \mathbf{R}_Q' + \mathbf{S}_Q' - \mathbf{T}_Q'] \end{aligned} \quad (8.68)$$



注:如果长方体的某条棱边与包含面 F 的平面发生碰撞,则要用位于平面 K_1 、 K_2 、 K_3 和 K_4 上的四个相互垂直的面裁剪连接两端点 C_1 和 C_2 的线段,以此检测此线段是否部分位于面 F 上

图 8.8 检测线段 C_1C_2 是否部分位于面 F 上

另外,裁剪四边形的平面的法线指向相应的长方体的外面,所以在裁剪过程中要进行求反操作。如果被四个平面裁剪后,四边形仍未被裁剪完,则一定发生了碰撞,碰撞检测运算停止。

完成长方体 Q 与长方体 P 的各个面之间的碰撞检测,下一步就要交换两个长方体的角色,即进行长方体 P 与长方体 Q 的各个面之间的碰撞检测。如果仍然没有发现碰撞,则需要检测一个长方体的棱边与另一个长方体的棱边在某一个点发生碰撞的可能性,这时两个长方体中没有一个边与另一个长方体的任何一个面平行。

当进行长方体 Q 与长方体 P 的各个面之间的碰撞检测时,记录下了长方体 Q 的哪一个顶点与哪一个包含长方体 P 的面的平面发生了碰撞,也包括 $t < 0$ 时发生碰撞的情况,反之亦然。如果将这一点作为算法中两个长方体边的交点,则与每个平面碰撞的点都位于长方体 P 中相应面的外面。此外,长方体 Q 的轴线与长方体 P 的轴线都不平行,故长方体 Q 不会与任何一个平面发生边或面碰撞。

对于长方体 P 的每一对平面 $L_1 = \langle \mathbf{N}_1, D_1 \rangle$ 和 $L_2 = \langle \mathbf{N}_2, D_2 \rangle$,考虑长方体 Q 中分别与

平面 L_1 和 L_2 在 C_1 和 C_2 发生碰撞的边。令 E_Q 表示长方体 Q 中连接 C_1 和 C_2 两点的边, E_P 表示长方体 P 中位于平面 L_1 和 L_2 的交线上的边, 如果棱边 E_Q 沿速度 V 的方向上的延伸与边 E_P 相交, 那么碰撞将发生在两条边相交的地方。 E_Q 沿速度 V 的方向上的延伸形成的平面的法线 N_3 为

$$N_3 = V \times (C_2 - C_1) \quad (8.69)$$

因此用 4D 坐标表示的平面向量 L_3 为

$$L_3 = \langle N_3, -N_3 \cdot C_1 \rangle \quad (8.70)$$

由方程(4.20)可以计算出三个平面 L_1 、 L_2 和 L_3 的交点 A 。这样就确保该点位于边 E_Q 在时间 $0 \leq t < 1$ 的延伸平面内, 定义向量 W 为

$$W = N_3 \times (C_2 - C_1) \quad (8.71)$$

如图 8.9 所示, W 位于边 E_Q 的延伸面上, 与边 E_Q 垂直, 并且远离速度 V 的方向。当式 8.72 成立时, 在边 E_Q 与 E_P 之间发生碰撞。

$$W \cdot (A - V - C_1) > 0 \quad (8.72)$$

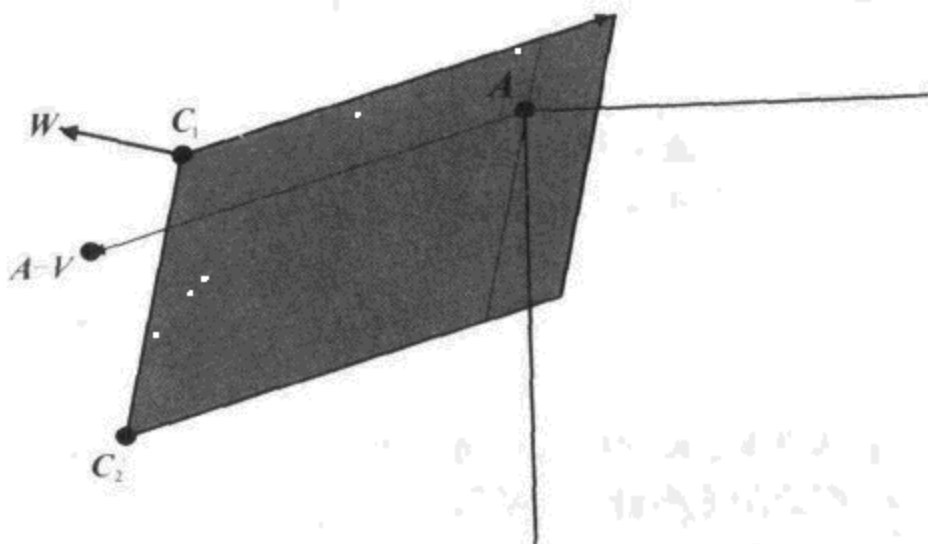


图 8.9 检测点 A 是否位于边 E_Q 的延伸面上

8.3 本章小结

球体与平面的碰撞

一个半径为 r 的球体, 其球心由 $t=0$ 时的 P_1 点运动到 $t=1$ 时的 P_2 点, 该球体与平面 $L = \langle N, D \rangle$ 在时刻 t

$$t = -\frac{L' \cdot P_1}{L' \cdot V}$$

发生碰撞, 其中 $L' = \langle N, D - r \rangle$ 。

长方体与平面的碰撞

一个棱边长度和方向分别为向量 R 、 S 和 T 的长方体, 其中心由 $t=0$ 时的 Q_1 点运动

到 $t=1$ 时的 Q_2 点,该长方体与平面 $L = \langle N, D \rangle$ 在时刻 t

$$t = -\frac{L' \cdot Q_1}{L' \cdot V}$$

发生碰撞,其中 $L' = \langle L, D - r_{\text{eff}} \rangle$, r_{eff} 为长方体的有效半径

$$r_{\text{eff}} = \frac{1}{2}(|R \cdot N| + |S \cdot N| + |T \cdot N|)$$

当长方体与平面在某一点发生碰撞时,与平面相切的顶点位置 C 为

$$C = Q(t) - \frac{1}{2}[\text{sgn}(R \cdot N)R + \text{sgn}(S \cdot N)S + \text{sgn}(T \cdot N)T]$$

其中 $Q(t) = Q_1 + t(Q_2 - Q_1)$ 。

两个球体之间的碰撞

一个半径为 r_P 球体,其球心由 $t=0$ 时的 P_1 点运动到 $t=1$ 时的 P_2 点。另一个半径为 r_Q 的球体,其球心由 $t=0$ 时的 Q_1 点运动到 $t=1$ 时的 Q_2 点。两个球体在时刻 t

$$t = \frac{-(A \cdot B) - \sqrt{(A \cdot B)^2 - B^2[A^2 - (r_P + r_Q)^2]}}{B^2}$$

发生碰撞,其中

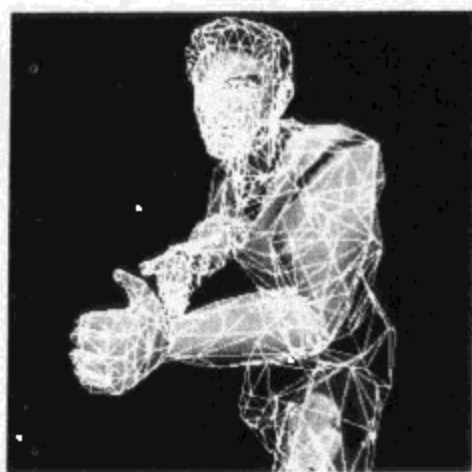
$$A = P_1 - Q_1$$

$$B = (P_2 - P_1) - (Q_2 - Q_1)$$

习 题

1. 计算一个半径为 2 米,球心在 $t=0$ 时位于原点,以恒定速度 $\langle 2, 0, 1 \rangle$ 运动的球体与平面 $x=10$ (米)之间发生碰撞的时间 t 。
2. 写一段检测两个球体在给定的时间段中是否发生碰撞的程序。程序的参数应为两个球体的初始位置和速度。如果发生碰撞,程序应当计算出碰撞发生的时间。
3. 编程实现球体-长方体碰撞检测的计算过程,并且并行执行射线的求交算法。该程序应该能够同时处理与表 8.2 中所列的四个平行圆柱面和表 8.3 所列的四个球面相交的情况。

第 9 章 多边形技术



本章将介绍与多边形模型的操作相关的技术,这些技术通常被用于实现游戏中的大量特殊效果,包括表面贴图、粒子系统和实时阴影等。

9.1 深度值修正

在许多游戏中,经常用到类似于墙上被火烧焦的痕迹或地上的足印这类特殊效果(在 9.2 一节中将介绍这一类特效的生成方法),这些都不是场景中原有的,而是在游戏过程中才出现的对象。表示这些特效的对象通常是粘贴在原来物体表面并与相应的多边形共面的多边形。由于两个多边形共面,造成在对一个多边形对应的像素进行渲染时,其像素的深度值很难与对另一个多边形对应的像素进行渲染时相同,最终导致原多边形的某些部分透过新多边形显示出来,而得到一个不想要的图案。

为了解决上述问题,对其中一个多边形在场景中的深度进行修正,同时保证该多边形在投影屏幕的坐标不变或者不改变它的纹理映射。大多数的三维图形系统提供了一些多边形修正函数,通过这些函数可以解决上述问题。然而,这些方法一般缺少好的控制,经常导致对每个像素点进行非常耗时的处理,这一节将介绍另外一种方法,也就是通过对投影矩阵的适当修改达到深度值修正的目的。

9.1.1 修改投影矩阵

首先来看标准的 OpenGL 透视投影矩阵在观察空间一点 $P = (P_x, P_y, P_z, 1)$ 处的效果。为了简化式(4.50)中给出的矩阵,假设视平截面以 z 轴为中心,则其左右平面在 $x = \pm n/e$ 处与其近平面相交,其上下平面在 $y = \pm an/e$ 处与其近平面相交,其中, e 为焦距, a 为图像的高宽比。引入点 P 与最近裁剪面的距离 n 以及点 P 与最远裁剪面的距离 f ,得到

$$\begin{bmatrix} e & 0 & 0 & 0 \\ 0 & \frac{e}{a} & 0 & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = \begin{bmatrix} eP_x \\ \left(\frac{e}{a}\right)P_y \\ -\frac{f+n}{f-n}P_z - \frac{2fn}{f-n} \\ -P_z \end{bmatrix} \quad (9.1)$$

为了完成投影的计算,需要用 w 坐标去除其余坐标项,此处 w 坐标的值为 $-P_z$,则投影点 P' 为

$$P' = \begin{bmatrix} -\frac{eP_x}{P_z} \\ -\frac{(e/a)P_y}{P_z} \\ \frac{f+n}{f-n} + \frac{2fn}{P_z(f-n)} \\ -P_z \end{bmatrix} \quad (9.2)$$

从方程(9.2)可以看出,保持 w 坐标值 $-P_z$ 不变,也会保证投影点的 x 和 y 坐标不变。依次类推,由于 z 坐标和 w 坐标值只受变换矩阵右下角 2×2 部分的影响,只注意矩阵中这一部分的数值。

对于一个无穷小 ϵ , 引入因子 $1 + \epsilon$ 可以在不影响 w 坐标值的前提下, 改变点 P 投影到 z 轴的坐标值, 如式 9.3 所示:

$$\begin{bmatrix} -(1 + \epsilon) \frac{f + n}{f - n} & -\frac{2fn}{f - n} \\ -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} P_z \\ 1 \end{bmatrix} = \begin{bmatrix} -(1 + \epsilon) \frac{f + n}{f - n} P_z - \frac{2fn}{f - n} \\ -P_z \end{bmatrix} \quad (9.3)$$

除 w 值后, 投影点的 z 坐标值为

$$P'_z = (1 + \epsilon) \frac{f + n}{f - n} + \frac{2fn}{P_z(f - n)} = \frac{f + n}{f - n} + \frac{2fn}{P_z(f - n)} + \epsilon \frac{f + n}{f - n} \quad (9.4)$$

与式(9.2)中 z 坐标值比较, 发现可以用一常量 $\epsilon \frac{f + n}{f - n}$ 来修正投影的深度值。

9.1.2 修正值的选择

由于 z 缓冲区的非线性特性, 式(9.4)中给出的常数修正值随着与观察点之间距离的远近而存在很大不同。尽管采用常数修正值在一些情况下应用得很好, 但是, 对于存在多种深度值的情况时, 采用常量修正值就不合适了。因此, 要根据观察空间中给定的不同深度 P_z 值选择合适的 ϵ 而得到一个相应的补充量 δ , 这样才能够准确地表示修正的对象。为了确定 ϵ 的计算公式, 将式(9.1)中的标准投影矩阵应用到一点, 该点的 z 坐标已进行了修正量为 δ 的修正

$$\begin{bmatrix} -\frac{f + n}{f - n} & -\frac{2fn}{f - n} \\ -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} P_z + \delta \\ 1 \end{bmatrix} = \begin{bmatrix} -\frac{f + n}{f - n} (P_z + \delta) - \frac{2fn}{f - n} \\ -(P_z + \delta) \end{bmatrix} \quad (9.5)$$

除 w 值后, 得到投影点的 z 坐标值为

$$\begin{aligned} P'_z &= \frac{f + n}{f - n} + \frac{2fn}{(P_z + \delta)(f - n)} \\ &= \frac{f + n}{f - n} + \frac{2fn}{P_z(f - n)} + \frac{2fn}{f - n} \left(\frac{1}{P_z + \delta} - \frac{1}{P_z} \right) \end{aligned} \quad (9.6)$$

带入式(9.4), 化简后得

$$\epsilon = -\frac{2fn}{f + n} \left(\frac{\delta}{P_z(P_z + \delta)} \right) \quad (9.7)$$

通过一个简单的试验就可以确定某个具体情况下 δ 的适当数值。另外, 因为 δ 为观察空间的修正量, 所以随着 P_z 的增大, δ 的有效性会降低。对于一个 m 位整数的深度缓冲区, 要确保式 9.8 成立:

$$|\epsilon| \geq \frac{1}{2^m - 1} \left(\frac{f + n}{f - n} \right) \quad (9.8)$$

否则, 由一个太小的 ϵ 值得出的修正值也会太小而不足够修改整数的深度值。将式(9.7)中 ϵ 表达式代入式(9.8), 求解得到 δ

$$\delta \geq \frac{kP_z^2}{1 - kP_z} \quad (9.9)$$

或

$$\delta \leq \frac{-kP_z^2}{1+kP_z} \quad (9.10)$$

其中, k 为常数, 且为

$$k = \frac{f-n}{2fn(2^m-1)} \quad (9.11)$$

式(9.9)给出了向观察点移动(通常情况)多边形时 δ 的最小有效值, 而式(9.10)则给出向远离观察点的方向移动多边形时 δ 的最大有效值。

9.1.3 应用举例

下面一段程序说明了式(9.3)中的投影矩阵在 OpenGL 中的实现方法。函数 LoadoffsetMatrix 的参数列表中除了含有传递给 OpenGL 函数 glFrustum 的 6 个参数, 还含有用于计算 ϵ 的 δ 和 P_z 两个参数。

【程序 9.1】 为了用公式(9.7)中得到的常数 ϵ 来修正深度值, 该段程序代码对 OpenGL 的投影矩阵进行了相应的修改。

```

Void LoadoffsetMatrix(GLdouble l, GLdouble r,
    GLdouble b, GLdouble t,
    GLdouble n, GLdouble f,
    GLfloat delta, GLfloat pz)
{
    GLfloat matrix[16];

    // 建立标准的透视投影
    glMatrixMode (GL_PROJECTION);
    glFrustum (l, r, b, t, n, f);

    // 获取投影矩阵
    glGetFloatv (GL_PROJECTION_MATRIX, matrix);

    // 计算  $\epsilon$ 
    GLfloat epsilon = -2.0F * f * n * delta /
        ((f + n) * pz * (pz + delta));

    // 修改矩阵中的 (3, 3) 元素
    matrix[10] *= 1.0F + epsilon;

    // 将修改后的投影矩阵返回 OpenGL
    glLoadMatrix (matrix);
}

```

9.2 贴图

墙面上被烧焦的痕迹和地面上的足印通常用构造贴图的方式实现,贴图与原来的表面一致,并用9.1一节中介绍的深度值修正技术进行渲染显示。将贴图附加于平面内部很简单,但如果将贴图附加在复杂曲面表示的对象表面或者地形表面时就会增加处理的难度。本节将介绍一个将贴图附加在任意形状的物体表面的常用方法,利用该方法可同时将贴图的边界裁剪得与物体表面一致。图9.1中所示的例子就是用本节介绍的方法实现的。

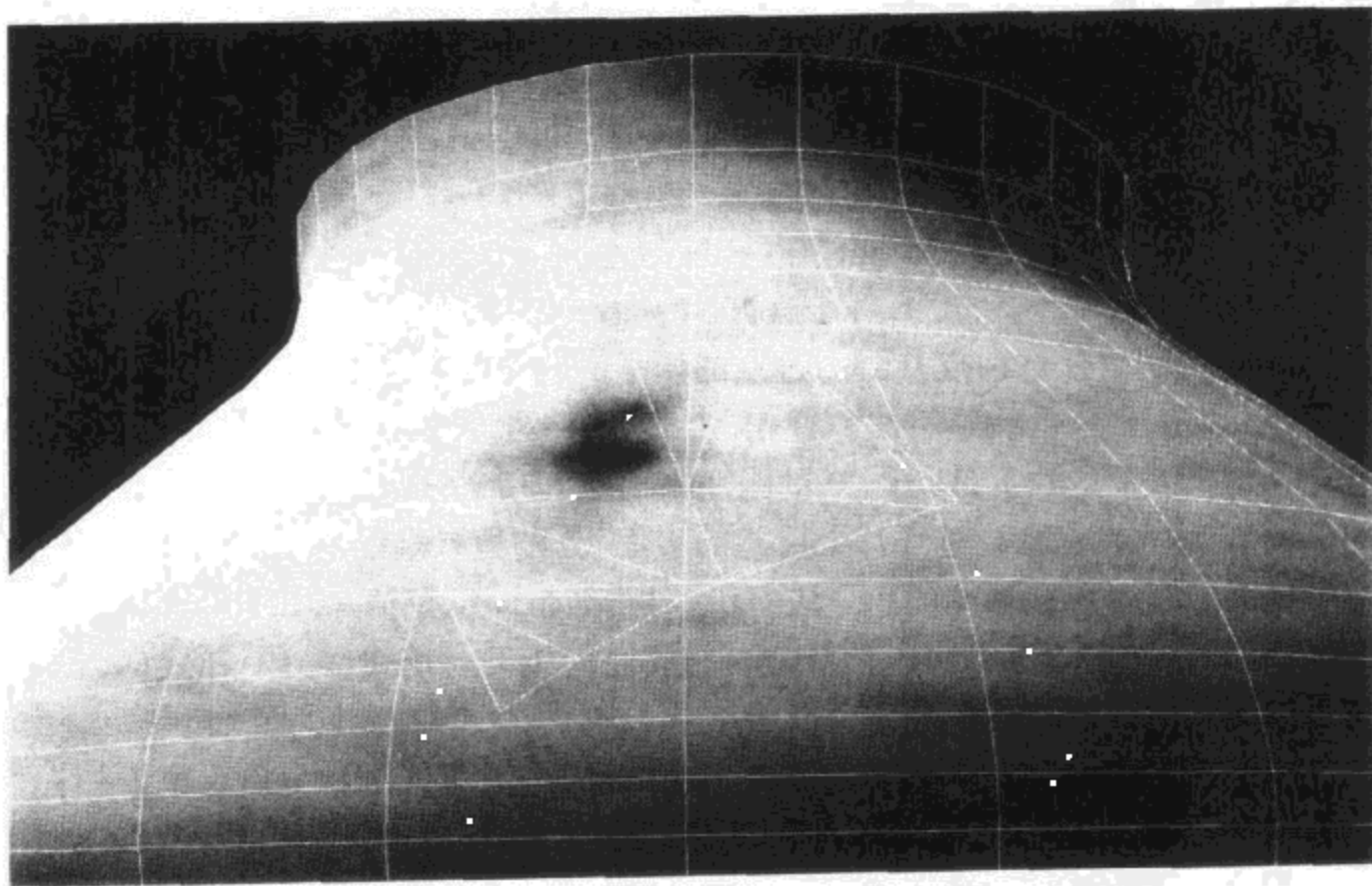


图 9.1 弯曲表面上的贴图

9.2.1 贴图网格

给定物体表面上一点 P 和在该点与物体表面垂直的单位法向量 N 。点 P 表示贴图中心,也就是子弹与物体表面的撞击点或者游戏中的人物在地面上的足迹的中心点。为了确定贴图的方向还必须确定点 P 处物体表面的单位切线方向 T ,如图9.2所示。

已知点 P 和单位方向向量 N 和 T ,可以获得一个在 P 点与物体表面几何体相切的平面。通过构造4个与法向量 N 平行的裁剪平面,可以将切平面裁剪成一个表示贴图区域的矩形。令 w 和 h 分

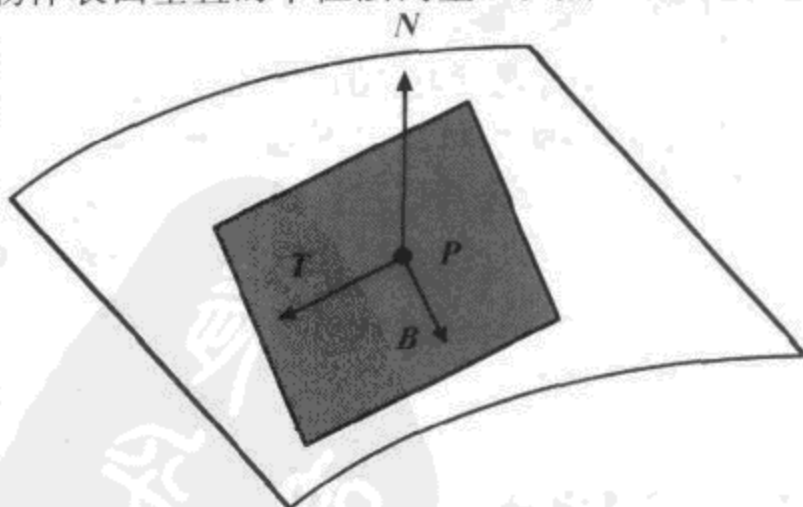


图 9.2 贴图参数

别表示贴图的长和宽,则 4 个裁剪平面的 4D 向量表达式分别为

$$\begin{aligned} \text{left} &= \left(\mathbf{T}, \frac{w}{2} - \mathbf{T} \cdot \mathbf{P} \right) \\ \text{right} &= \left(-\mathbf{T}, -\frac{w}{2} + \mathbf{T} \cdot \mathbf{P} \right) \\ \text{bottom} &= \left(\mathbf{B}, \frac{h}{2} - \mathbf{B} \cdot \mathbf{P} \right) \\ \text{top} &= \left(-\mathbf{B}, -\frac{h}{2} + \mathbf{B} \cdot \mathbf{P} \right) \end{aligned} \quad (9.12)$$

其中 $\mathbf{B} = \mathbf{N} \times \mathbf{T}$ 。用这 4 个裁剪平面对贴图附近的物体表面进行裁剪可以构造出贴图的三角网格。另外还要对点 \mathbf{P} 对应的前平面和后平面进行裁剪,这是为了避免那些位于 4 个裁剪平面构成的边界内部而在点 \mathbf{P} 前面或者后面非常远的物体的表面被裁剪掉。前平面与后平面的 4D 向量表达式分别为

$$\begin{aligned} \text{front} &= (-\mathbf{N}, d + \mathbf{N} \cdot \mathbf{P}) \\ \text{back} &= (\mathbf{N}, d - \mathbf{N} \cdot \mathbf{P}) \end{aligned} \quad (9.13)$$

其中 d 表示在贴图上任意顶点 \mathbf{P} 到通过点的切平面的最远距离。

下面介绍贴图网格的构造算法。首先,辨别贴图将会附加在场景中的哪些表面上,这可以通过计算每个表面的边界体与点 \mathbf{P} 的距离来确定,即那些边界体与点 \mathbf{P} 的距离小于某个数值的表面即为所求表面。对于每个可能的表面,分别检查其网格中的每个三角形,令 \mathbf{M} 为网格中某个三角形所在平面的单位法向量,对于某个固定正值 ϵ ,如果 $\mathbf{N} \cdot \mathbf{M} < \epsilon$,则排除该三角形,因为这些三角形的法向量与贴图的法向量相反。用式(9.12)和式(9.13)对剩下的三角形进行裁剪形成一个新的三角形网格。

当一个三角形与任意一个裁剪平面相交而被裁剪时,其法向量与顶点位置一样都通过插值计算得出,这样可使后来计算出的三角形新顶点的颜色能够反应每个顶点法向量与贴图的法向量之间的夹角。最终可以获得贴图纹理相对三角形的方向平滑过渡的效果。我们用下式为每个顶点定义一个 alpha 值:

$$\text{alpha} = \frac{\frac{\mathbf{N} \cdot \mathbf{R}}{\|\mathbf{R}\|} - \epsilon}{1 - \epsilon} \quad (9.14)$$

其中, \mathbf{R} 为该顶点的法向量(由于插值可能为非规范化向量)。这样就将点积区间 $[\epsilon, 1]$ 映射到 alpha 的取值区间 $[0, 1]$ 。

经过上述步骤最终获得的三角网格就是贴图网格。计算该网格的每个顶点与经过点 \mathbf{P} 并且法向量分别为 \mathbf{T} 和 \mathbf{B} 的平面之间的距离,可以得出贴图网格的纹理映射坐标。令 \mathbf{Q} 为贴图网格中一个顶点的位置,则纹理坐标 s 和 t 分别为

$$\begin{aligned} s &= \frac{\mathbf{T} \cdot (\mathbf{Q} - \mathbf{P})}{w} + \frac{1}{2} \\ t &= \frac{\mathbf{B} \cdot (\mathbf{Q} - \mathbf{P})}{h} + \frac{1}{2} \end{aligned} \quad (9.15)$$

9.2.2 多边形裁剪

物体表面中要进行贴图部分中的三角形通常被当作凸多边形来处理,这些三角形依次

与9.2.1节中介绍的6个裁剪平面进行裁剪。已知用平面对一个含有 n 个顶点的凸多边形进行裁剪可产生一个最多含有 $n+1$ 个顶点的凸多边形。因此,用6个裁剪平面对三角形进行裁剪产生一个最多含有9个顶点的多边形。裁剪完成后,要对新的多边形进行三角剖分,然后作为三角形扇添加到贴图的三角形网格中。

为了用任意平面对凸多边形进行裁剪,首先将多边形的所有顶点分成两类,即裁剪平面负半空间中的顶点和裁剪平面正半空间或者裁剪平面上的顶点。这与7.4.1一节中裁剪门的分类不同,没有将位于裁剪平面上的点单独列为一类。如果多边形的所有顶点都在平面的负半空间,那么这个多边形就可以被删除,否则,遍历多边形中的每对相邻顶点,检查它们对应的多边形边是否与平面相交,如图9.3所示,如果相交,交点就成为多边形的新顶点,而位于裁剪平面负半空间的顶点就被删去。

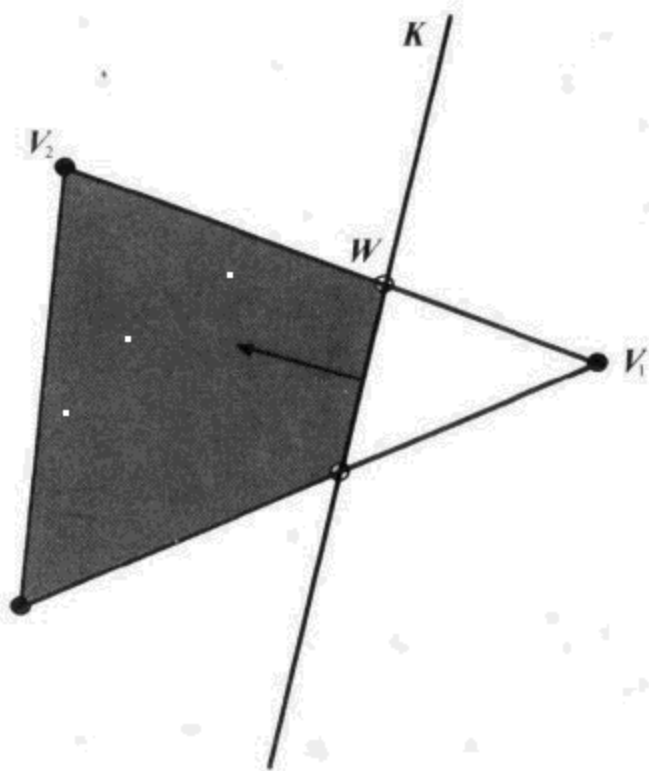


图9.3 平面裁剪多边形

假设顶点在 V_1 在裁剪平面 K 的正半空间,顶点 V_2 在 K 的负半空间,则 V_1 和 V_2 的连线上一点 W 可表示为

$$W(t) = V_1 + t(V_2 - V_1) \quad (9.16)$$

其中,参数 t 满足 $0 \leq t \leq 1$ 。当 $K \cdot W(t) = 0$ 时, t 为

$$t = \frac{K \cdot V_1}{K \cdot (V_1 - V_2)} \quad (9.17)$$

注意, $V_1 - V_2$ 差的 w -坐标为0。将 t 值代回式9.16,就得到新顶点 W 的表达式。

9.3 广告牌技术

许多特殊效果是通过将二维纹理映射到一个平面多边形上实现的,该多边形始终面向观察点的位置。这种技术称为广告牌技术,是使平面物体产生立体感的有效方法。这一节将探讨在不同情况下计算广告牌顶点的方法。

9.3.1 无约束四边形

无约束四边形是指具有四条边的矩形多边形,它可以沿任意方向旋转。无约束四边形可以用来创造类似于粒子系统、飞行器尾部烟雾以及闪烁的光环等特殊效果。

通过将四边形的顶点限制在一个与观察视线垂直的平面上,可以构造一个无约束四边形广告牌。令单位向量 \mathbf{R} 和 \mathbf{U} 分别表示世界坐标系中观察视图的右向和顶向,也就是观察坐标系的 x - 和 y - 的方向。四边形广告牌可以通过以下参数来定义:

- (1) 四边形的中心,即世界坐标系中的点 \mathbf{P} ;
- (2) 四边形的宽 w 和高 h 。它们能够随着时间的变化而变化,从而产生广告牌的缩放效果;
- (3) 四边形相对观察方向的旋转角度 θ 。 θ 能够随着时间的变化而变化,从而产生广告牌的旋转效果。如果 θ 为常数,则广告牌与观察点一起沿观察方向旋转。

根据上述参数,定义向量 \mathbf{X} 和 \mathbf{Y}

$$\mathbf{X} = \left(\frac{w}{2}\cos\theta\right)\mathbf{R} + \left(\frac{w}{2}\sin\theta\right)\mathbf{U} \quad \mathbf{Y} = \left(-\frac{h}{2}\sin\theta\right)\mathbf{R} + \left(\frac{h}{2}\cos\theta\right)\mathbf{U} \quad (9.18)$$

旋转角度 θ 的数值一般量化为一些可能的角度值,然后通过一个查找表来求其正弦与余弦值。当然,当 $\theta=0$ 时,向量 \mathbf{X} 和 \mathbf{Y} 的表达式简化为

$$\begin{aligned} \mathbf{X} &= \frac{w}{2}\mathbf{R} \\ \mathbf{Y} &= \frac{h}{2}\mathbf{U} \end{aligned} \quad (9.19)$$

如图 9.4 所示,四边形广告牌的四个顶点 \mathbf{Q}_1 、 \mathbf{Q}_2 、 \mathbf{Q}_3 和 \mathbf{Q}_4 分别为

$$\begin{aligned} \mathbf{Q}_1 &= \mathbf{P} + \mathbf{X} + \mathbf{Y} & \mathbf{Q}_2 &= \mathbf{P} - \mathbf{X} + \mathbf{Y} \\ \mathbf{Q}_3 &= \mathbf{P} - \mathbf{X} - \mathbf{Y} & \mathbf{Q}_4 &= \mathbf{P} + \mathbf{X} - \mathbf{Y} \end{aligned} \quad (9.20)$$

这些顶点以逆时针方向排列,保证了相应的四边形总是面向观察点。相应的二维纹理映射坐标为

$$\begin{aligned} \langle s_1, t_1 \rangle &= \langle 1, 1 \rangle & \langle s_2, t_2 \rangle &= \langle 1, 0 \rangle \\ \langle s_3, t_3 \rangle &= \langle 0, 0 \rangle & \langle s_4, t_4 \rangle &= \langle 1, 0 \rangle \end{aligned} \quad (9.21)$$

由式(9.18)中的向量 \mathbf{X} 和 \mathbf{Y} 得出的顶点构成了四边形广告牌,该广告牌始终与观察平面平行,如图 9.5 所示,这和垂直于四边形中心到观察位置的实际方向的平面有明显不同。当要渲染成百上千的微小粒子时,用式(9.18)可以获得需要的效率。但是,对于尺寸较大的广告牌,如果该广告牌面对实际的观察位置,就会获得比面对观察平面更好的视觉效果。

为了使广告牌能够面对观察点的位置,这里提供了一个计算向量 \mathbf{X} 和 \mathbf{Y} 公式,该公式的计算量要比式(9.18)大得多。令向量 \mathbf{C} 表示世界坐标系中观察点的位置,假设四边形的中心 \mathbf{P} 不位于经过 \mathbf{C} 点并沿向量 \mathbf{U} 的方向延伸的直线上,进行如下运算:

$$\begin{aligned} \mathbf{Z} &= \frac{\mathbf{C} - \mathbf{P}}{\|\mathbf{C} - \mathbf{P}\|} \\ \mathbf{A} &= \frac{\mathbf{U} \times \mathbf{Z}}{\|\mathbf{U} \times \mathbf{Z}\|} \\ \mathbf{B} &= \mathbf{Z} \times \mathbf{A} \end{aligned} \quad (9.22)$$

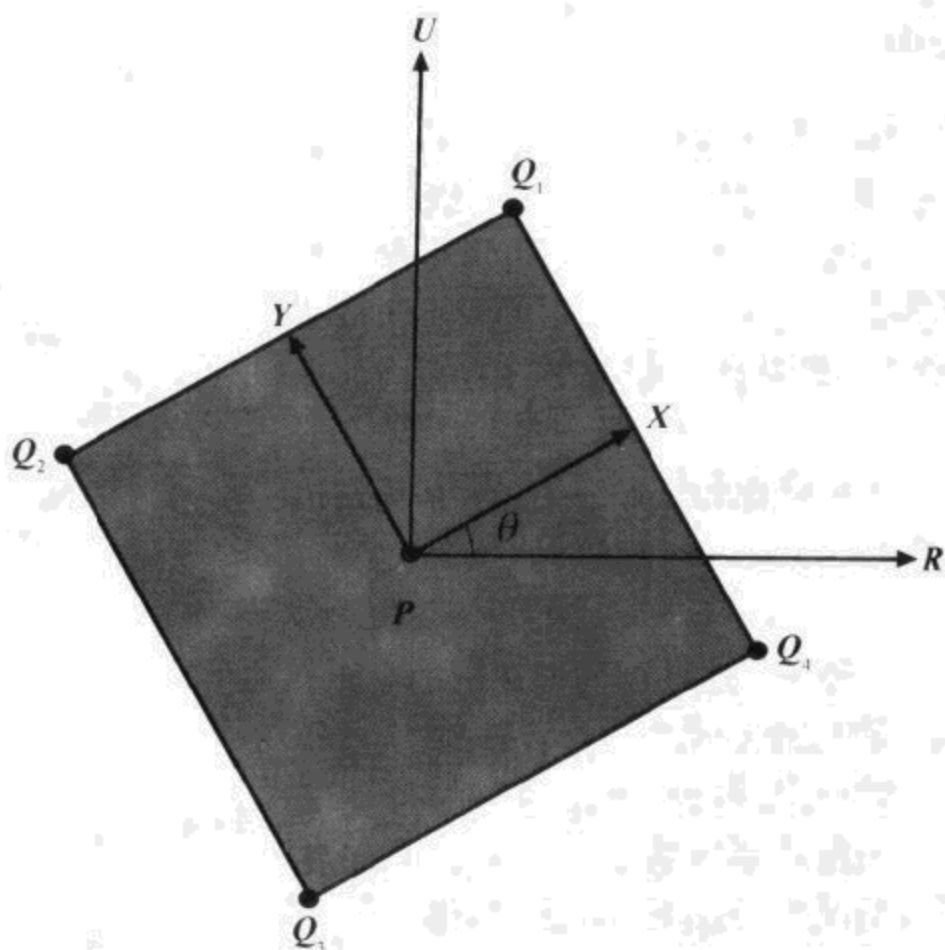


图 9.4 计算无约束四边形公告牌的顶点

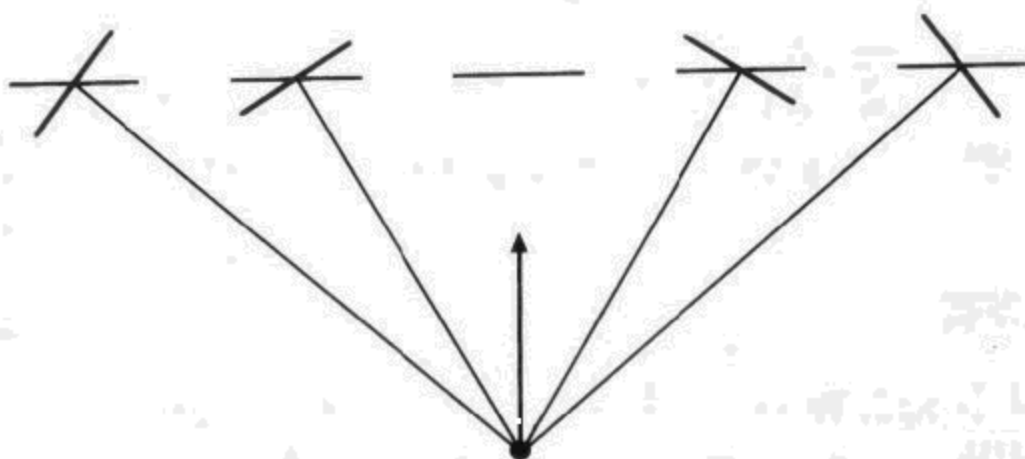


图 9.5 面对观察平面的广告牌与面对观察位置的广告牌存在明显不同

其中,向量 Z 是由四边形中心指向观察位置的单位向量,计算向量 Z 与 U 的叉乘可得位于广告牌平面的正交向量 A 。如果 $U \times A$ 的值接近 0,那么用以下公式替换式(9.22)中向量 A 和 B 的计算公式

$$B = \frac{Z \times R}{\|Z \times R\|}$$

$$A = B \times Z \tag{9.23}$$

向量 A 和 B 为一对正交向量,用它们可以表示向量 X 和 Y

$$X = \left(\frac{w}{2} \cos\theta\right)A + \left(\frac{w}{2} \sin\theta\right)B \quad Y = \left(-\frac{h}{2} \sin\theta\right)A + \left(\frac{h}{2} \cos\theta\right)B \tag{9.24}$$

代入式(9.20)得广告牌四边形的 4 个顶点。

9.3.2 约束四边形

下面介绍如何约束一个四边形,使它只能围绕 z 坐标轴旋转的方法。这种四边形可用于产生火炬的火焰纹理,这种情况下,火焰总是向上的,但是四边形也总是被旋转以保持面向观察点。只要不从四边形的正上方或者正下方观察,用这种方法产生的火焰效果具有很强的立体感。

假设世界坐标系中观察点的位置为 C ,点 P 为四边形的中心,则向量 X 为

$$\mathbf{X} = \langle P_y - C_y, C_x - P_x, 0 \rangle \quad (9.25)$$

如图 9.6 所示,向量 X 是将观察点 C 与四边形的中心 P 的差投影到 xy 平面,然后绕 z 轴逆时针旋转 90° 而计算出来的。当 $\|\mathbf{X}\| = 0$ 时,观察点位于四边形的正上方或正下方,这种情况下,只能看到四边形的一条边,因此不进行显示计算。其他情况要计算四边形的四个顶点 Q_1 、 Q_2 、 Q_3 和 Q_4 为

$$\begin{aligned} Q_1 &= P + \frac{w}{2} \frac{\mathbf{X}}{\|\mathbf{X}\|} + \langle 0, 0, \frac{h}{2} \rangle & Q_2 &= P - \frac{w}{2} \frac{\mathbf{X}}{\|\mathbf{X}\|} + \langle 0, 0, \frac{h}{2} \rangle \\ Q_3 &= P - \frac{w}{2} \frac{\mathbf{X}}{\|\mathbf{X}\|} - \langle 0, 0, \frac{h}{2} \rangle & Q_4 &= P + \frac{w}{2} \frac{\mathbf{X}}{\|\mathbf{X}\|} - \langle 0, 0, \frac{h}{2} \rangle \end{aligned} \quad (9.26)$$

纹理映射坐标的计算方法与公式(9.21)所描述的无约束四边形的纹理映射坐标的计算方法一样。

9.3.3 四边形带

如图 9.7 所示,通过为含有 N 个顶点 P_1, P_2, \dots, P_N 的折线定义宽度 r ,可以构造一个与该折线一致的四边形带。四边形带可用于实现一连串闪电的特殊效果,而这些闪电的路径由一系列点组成。四边形带的另一个应用是渲染粒子的运动模糊效果,这时要计算出前一帧中粒子的位置与当前粒子的位置之间的一些中间位置,这些位置一起构成四边形带中折线的顶点。

对于折线中的每一个点 P_i ,可以构造出两个四边形顶点,这两个顶点分别位于与点 P_i 距离为 r 的地方。这两个顶点与点 P_i 所在的直线应该与点 P_i 到观察位置的方向以及折线在点 P_i 处的切线互相正交。点 P_i 到观察位置的单位方向向量 Z_i 为

$$\mathbf{Z}_i = \frac{\mathbf{C} - \mathbf{P}_i}{\|\mathbf{C} - \mathbf{P}_i\|} \quad (9.27)$$

其中, C 为观察点的位置,点 P_i 处折线的单位切向量 T_i 为

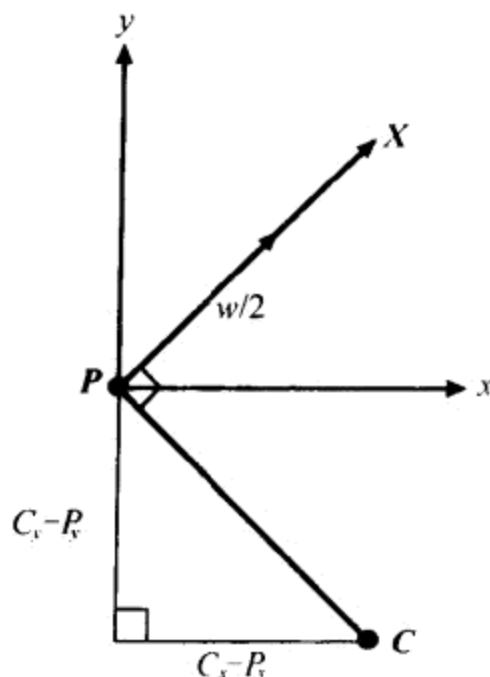


图 9.6 只能绕 z 坐标轴旋转的四边形广告牌顶点的计算方法

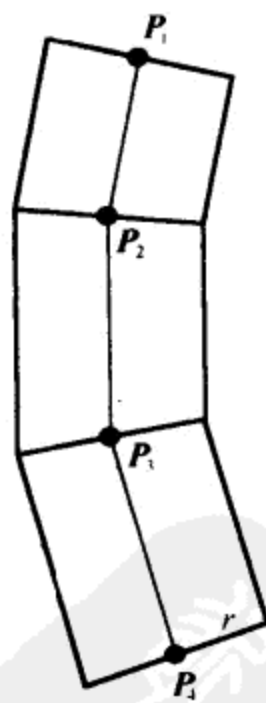


图 9.7 半径为 r 沿多边形分布四边形带

$$T_i = \frac{P_{i+1} - P_{i-1}}{\|P_{i+1} - P_{i-1}\|} \quad (9.28)$$

当 P_i 为折线端点时,相应的单位切向量为

$$\begin{aligned} T_1 &= \frac{P_2 - P_1}{\|P_2 - P_1\|} \\ T_N &= \frac{P_N - P_{N-1}}{\|P_N - P_{N-1}\|} \end{aligned} \quad (9.29)$$

与点 P_i 对应的两个四边形顶点 G_i 和 H_i 为

$$\begin{aligned} G_i &= P_i + r(T_i \times Z_i) \\ H_i &= P_i - r(T_i \times Z_i) \end{aligned} \quad (9.30)$$

四边形带的每一条边由顶点 $G_1, H_1, G_2, H_2, \dots, G_N, H_N$ 构成,这些边分别垂直于观察方向。图 9.8 所示的就是用这一技术生成的闪电的特殊效果。

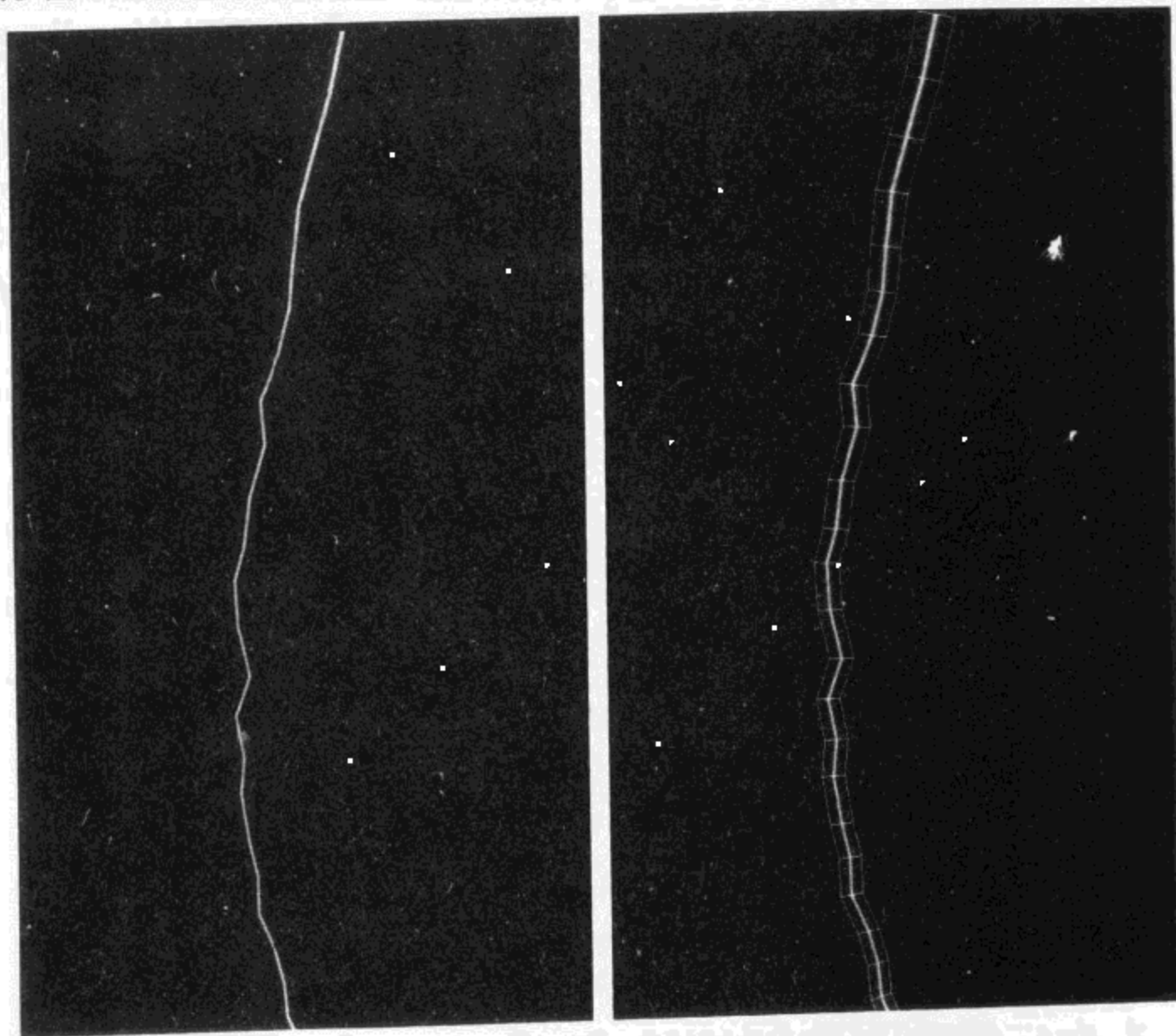


图 9.8 用于闪电效果的多边形带

9.4 模板阴影

本节将介绍用于准确地生成任意闭合三角形网格的阴影的技术。所谓闭合的三角形网格是指该网格中的每个边都被偶数个三角形共享。这种要求不允许三角形网格中存在漏

洞,通过它可以看到模型表面的内部。

首先构造三角形网格中所有边的列表,对于列表中的每条边来说只有两个三角形共享该边,对于由多个三角形共享的边,要按两个三角形共享一条边的规则复制该边。当生成三角形网格相对于某个光源的阴影时,须确定每个三角形是面向光源还是背离光源。共享一个边的两个三角形中,如果一个面向光源而另一个背向光源,则该边被认为是网格阴影轮廓的一部分。轮廓中的任意一边沿着光线方向延伸形成一个四边形,与轮廓中的边对应的一组四边形将围成一个空间区域,称为阴影体。

通过专门的模板缓冲区操作,阴影体的渲染可以通过在阴影产生的准确位置将阴影写入模板缓冲区的方法实现。下面将详细介绍这种方法。

9.4.1 构造边表

假设一个三角形网格的索引结构中含有一个由 N 个顶点 V_1, V_2, \dots, V_N 组成的顶点数组和一个由 M 个三角形 T_1, T_2, \dots, T_M 组成的三角形数组。每个三角形结构中只含有构成该三角形的三个顶点的整数索引 i_1, i_2 和 i_3 。如果在循环链 $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$ 中下标 p 在 q 之前,则索引 i_p 也在 i_q 之前,例如, i_2 在 i_3 前, i_3 在 i_1 前,而 i_2 不在 i_1 前。

索引 i_1, i_2 和 i_3 的排列顺序必须保证它们对应的顶点 V_{i_1}, V_{i_2} 和 V_{i_3} 绕三角形的法向量逆时针排列。假设两个三角形的公共边的两个端点分别是 V_a 和 V_b ,如图 9.9 所示,按一致的顶点排列规则使这两个三角形中顶点的排列具有以下特点,对于一个三角形, V_a 的索引位于 V_b 之前,而对于另一个三角形来说, V_a 的索引则位于 V_b 之后。

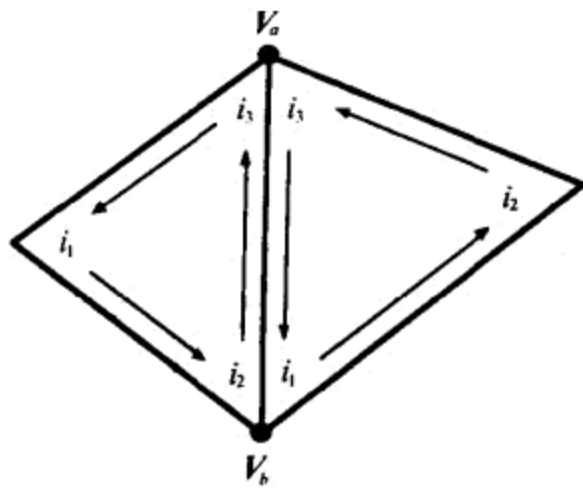


图 9.9 顶点排列规则

对三角形网格的三角形列表进行一次遍历就可确定每个三角形网格中的边。对于顶点索引为 i_1, i_2 和 i_3 的三角形,为满足 $i_1 < i_2, i_2 < i_3$ 和 $i_3 < i_1$ 三个条件的顶点分别构造三个边记录,并将三角形索引保存在边记录中。通过这一过程,可以为共享两个顶点 V_a 和 V_b 的每对三角形正好只创建了一条边,对于多对三角形共享同一条边的情况,则只要复制该边的结构就可以了。

一旦确定了三角形网格中的所有边,对三角形列表进行第二次遍历以寻找每个共享边的另一个三角形。首先确定满足 $i_1 > i_2, i_2 > i_3$ 和 $i_3 > i_1$ 的三角形,然后将该三角形的索引保存到含有相同顶点索引并且第二个三角形索引为空的边结构中。这种方法也适用于被复制的边结构。程序 9.2 中的代码说明了边表的构造方法。

【程序 9.2】 利用三角形的索引数组构造一个边记录数组,边记录中含有指向共享该边的两个三角形的索引结构。程序的返回值为写入数组 edgeArray 中的边的数量。

```

Struct Edge
{
    unsigned short  vertexIndex[2];
    unsigned short  triangleIndex[2];
};

Struct Triangle
{
    unsigned short  index[3];
};

long BuildEdges long triangleCount,const Triangle *triangleArray,
    Edge **edgeArray
{
    // 为边表分配存储空间
    edgeArray=new  Edge[triangleCount *3],

    long  edge Count = *edgeArray;
    Edge  *edge = *edgeArray;

    // 第一次遍历,发现边
    Const Triangle *triangle =triangleArray;
    For  long  a=0; a,triangleCount ;a++
    {
        long i1 =triangle → index[0]
            long i2 = triangle→  index[1];
            long i3 = triangle →  index[2];.

        if (i1<i2)
        {
            edge→vertexIndex[0] =i1;
            edge→vertexIndex[1]=i2;
            edge→triangleIndex[0]=a;
            edgevtriangleIndex[1]=-1;
            edgeCount++;
            edge++;
        }
    }
}

```

```
        if (i2<i3)
        {
            edge->vertexIndex[0] =i2;
            edge->vertexIndex[1]=i3;
            edge->triangleIndex[0]=a;
            edge->triangleIndex[1]=-1;
            edgeCount++;
            edge++;
        }

        if(i3<i1)
        {
            edge->vertexIndex[0] =i3;
            edge->vertexIndex[1]=i1;
            edge->triangleIndex[0]=a;
            edge->triangleIndex[1]=-1;
            edgeCount++;
            edge++;
        }

        triangle++;
    }
}
```

// 第二次遍历, 将三角形索引添加到边记录中

```
triangle = triangleArray;
for (long a = 0;a<triangleCount;a++)
{
    long i1 = triangle->index[0];
    long i2 = triangle->index[1];
    long i3 = triangle->index[2];

    if (i1>i2)
    {
        edge = *edgeArray;
        for (long b=0;b<edgeCount;b++)
        {
            if ((edge->vertexIndex[0]==i2)&&
                edge->vertexIndex[1]==i1)&&
                (edge->triangleIndex[1]==-1))
            {
```

```
        {
            edge->triangleIndex[1]=a;
            break;
        }
        edge++;
    }
}

if (i2>i3)
{
    edge = *edgeArray;
    for (long b=0;b<edgeCount;b++)
    {
        if ((edge->vertexIndex[0]==i3)&&
            (edge->vertexIndex[1]==i2)&&
            (edge->triangleIndex[1]==-1))
        {
            edge->triangleIndex[1]=a;
            break;
        }
        edge++;
    }
}

if (i3>i1)
{
    edge = *edgeArray;
    for (long b=0;b<edgeCount;b++)
    {
        if ((edge->vertexIndex[0]==i1)&&
            (edge->vertexIndex[1]==i3)&&
            (edge->triangleIndex[1]==-1))
        {
            edge->triangleIndex[1]=a;
            break;
        }
        edge++;
    }
}
```

```

        triangle++
    }
    return (edgeCount)
}

```

9.4.2 阴影渲染

利用三角网格的边表,就可以确定模型相对于一个已知光源的轮廓。首先要计算出每个三角形的法向量 \mathbf{N} 与光源方向 \mathbf{D} 的点积。对于顶点索引为 i_1, i_2 和 i_3 的三角形,由于顶点绕法线方向逆时针排列,则向外的法向量(不一定为规范化向量)为

$$\mathbf{N} = (\mathbf{V}_{i_2} - \mathbf{V}_{i_1}) \cdot (\mathbf{V}_{i_3} - \mathbf{V}_{i_1}) \quad (9.31)$$

如果 $\mathbf{N} \cdot \mathbf{D} > 0$, 则三角形面向光源, 否则, 三角形背向光源。

通过遍历边表中的每个边并检查共享该边两个面的方向就可确定模型的轮廓。共享一条边的两个三角形中, 如果一个三角形面向光源而另一个三角形背向光源, 则公共边为模型轮廓的一部分。当模型轮廓被确定后, 将其中的每一条边沿光线方向延伸形成相应的四边形, 并将这些四边形存储到渲染队列中(四边形的存储顺序不一定是连续的)。四边形必须沿逆时针方向排列, 因此必须按正确的顺序排列顶点。假设两个端点分别为 \mathbf{V}_a 和 \mathbf{V}_b 的一条轮廓边, 如果该边的第一个邻接三角形面向光源, 那么该边延伸出的四边形的顶点为

$$\begin{aligned} Q_1 &= \mathbf{V}_b & Q_2 &= \mathbf{V}_a \\ Q_3 &= \mathbf{V}_a - m\mathbf{D} & Q_4 &= \mathbf{V}_b - m\mathbf{D} \end{aligned} \quad (9.32)$$

其中, m 是阴影的长度, 并且假设 \mathbf{D} 为单位长度的向量。相反, 如果第一个三角形背向光源, 那么相应的四边形的顶点为

$$\begin{aligned} Q_1 &= \mathbf{V}_a & Q_2 &= \mathbf{V}_b \\ Q_3 &= \mathbf{V}_b - m\mathbf{D} & Q_4 &= \mathbf{V}_a - m\mathbf{D} \end{aligned} \quad (9.33)$$

顶点 Q_3 和 Q_4 可能是该四边形与一个平面 L 的交点, 这时令

$$m_a = \frac{\mathbf{L} \cdot \mathbf{V}_a}{\mathbf{L} \cdot \mathbf{D}} \quad m_b = \frac{\mathbf{L} \cdot \mathbf{V}_b}{\mathbf{L} \cdot \mathbf{D}} \quad (9.34)$$

并用 $\mathbf{V}_a - m_a\mathbf{D}$ 替换 $\mathbf{V}_a - m\mathbf{D}$, 用 $\mathbf{V}_b - m_b\mathbf{D}$ 替换 $\mathbf{V}_b - m\mathbf{D}$ 就可求出这两个顶点。这样做可以保证阴影不会穿过墙面和房间的地板。

9.4.3 实现

现在假设已经获得四边形数组, 这些四边形共同组成了三角形网格的阴影体的边界。利用模板操作, 可以计算出这个阴影。将阴影体渲染两次就可计算出实际的阴影, 第一次只渲染四边形的前向面, 而第二次只渲染四边形的后向面。在第一次渲染中, 当 z 缓冲区测试通过时, 就增加模板缓冲区的值, 在第二次渲染中, 当 z 缓冲区测试通过时, 减少模板缓冲区的值。对于阴影中的任一像素, 在第二次渲染中 z 缓冲区测试未通过, 因此模板缓冲区中留下一个为 1 的值。程序 9.3 说明了 OpenGL 中实现这些操作的方法。

【程序 9.3】 本程序说明了用 OpenGL 编程实现阴影渲染的一些必要操作。

```
// 禁止向颜色缓冲区和 z 缓冲区的写操作
glColorMask(GL_FALSE, GL_FALSE, GL_FALSE);
glDepthMask(GL_FALSE);

// 设置模板测试总是成功
glStencilFunc(GL_ALWAYS, 0, 255);

// 渲染前向面
glCullFace(GL_BACK);
glStencilOp(GL_KEEP, GL_KEEP, GL_INCR);
glDrawArrays(...);

// 渲染后向面
glCullface(GL_FRONT);
glStencilOP(GL_KEEP, GL_KEEP, GL_INCR);
glDrawArrays(...);

// 恢复向颜色缓冲区和 z 缓冲区的写操作
glColorMask(GL_TRUE, GL_TRUE, GL_TRUE, GL_TRUE);
glDepthMask(GL_TRUE);
glCullface(GL_BACK);
```

一旦将测试结果写入模板缓冲区,模板缓冲区的值不为0的地方即为阴影,通过在整个屏幕显示一个含有 alpha 通道的四边形就可使阴影区域变黑而形成阴影的效果。在 OpenGL 中,通过以下函数调用就可实现模板测试,

```
glStencilOP(GL_KEEP, GL_KEEP, GL_KEEP);
glStencifunc(GL_NOTEQUAL, 0, 255);
```

图 9.10 所示的就是这种操作的结果,其中人物的阴影体就是用本节中的方法确定的。另外,图中还显示了与该阴影体对应的轮廓边。

9.5 多边形简化

当一个包含大量三角形的模型离观察点很远时,其中的一些三角形对模型最终的渲染图像没有明显影响。随模型与观察点之间距离的增大,适当减少要渲染的三角形的数量可以降低处理的计算量以及减少传送给图形硬件的数据量。

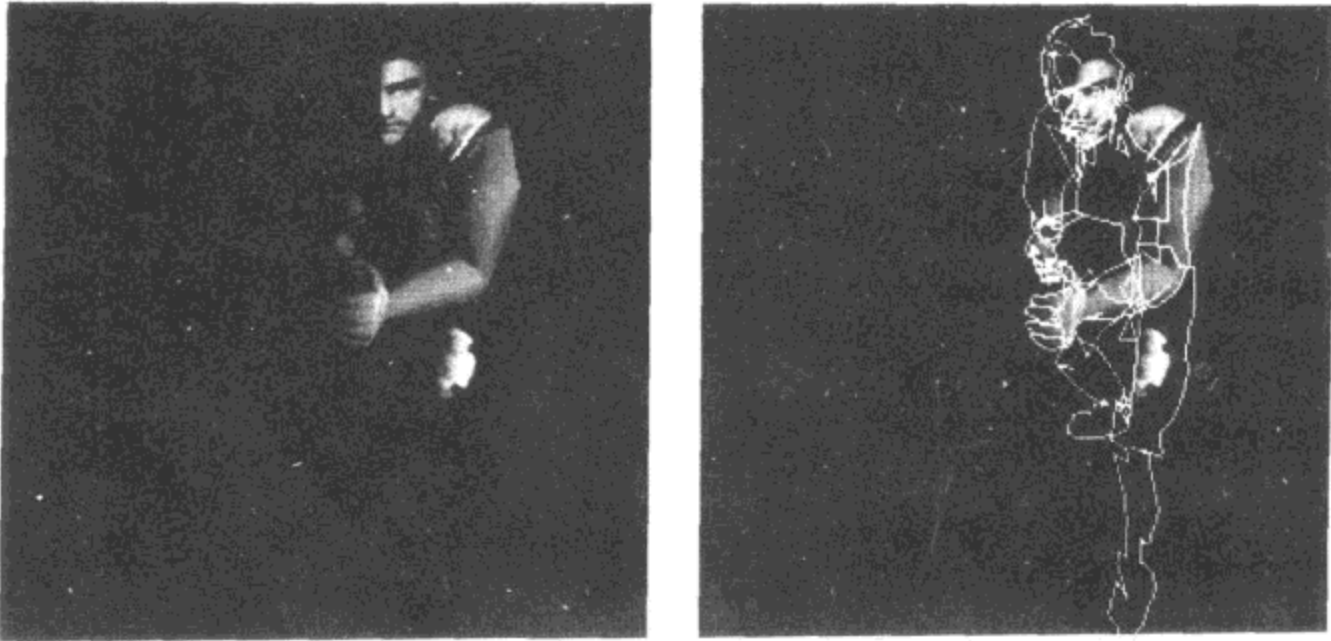


图 9.10 通过将右图中所示的轮廓边延伸并应用程序 9.3 中模板操作构造人物的阴影

减少网格中三角形数量的常用方法是边压缩方法。这种方法的主要目的就是寻找三角网格中的一些边,如果删去这些边不会引起三角网格形状的巨大改变,就将其从三角网格中删除。删除这些边的过程被称为边压缩,它是通过合并边的两个端点来实现的,如图 9.11 所示,其中一个端点保持不变,而另一个端点移到第一个端点的位置。这样,根据两个端点的取舍,边压缩可按两种方式进行。共享压缩边的两个三角形将被删除,而留下的漏洞则通过将共享移动端点的三角形进行延伸来弥补。当然,由于两个端点占据了相同的位置,可以删除被移动的端点,因此,一条边的压缩将会删除网格中的两个三角形、一条边和一个顶点。

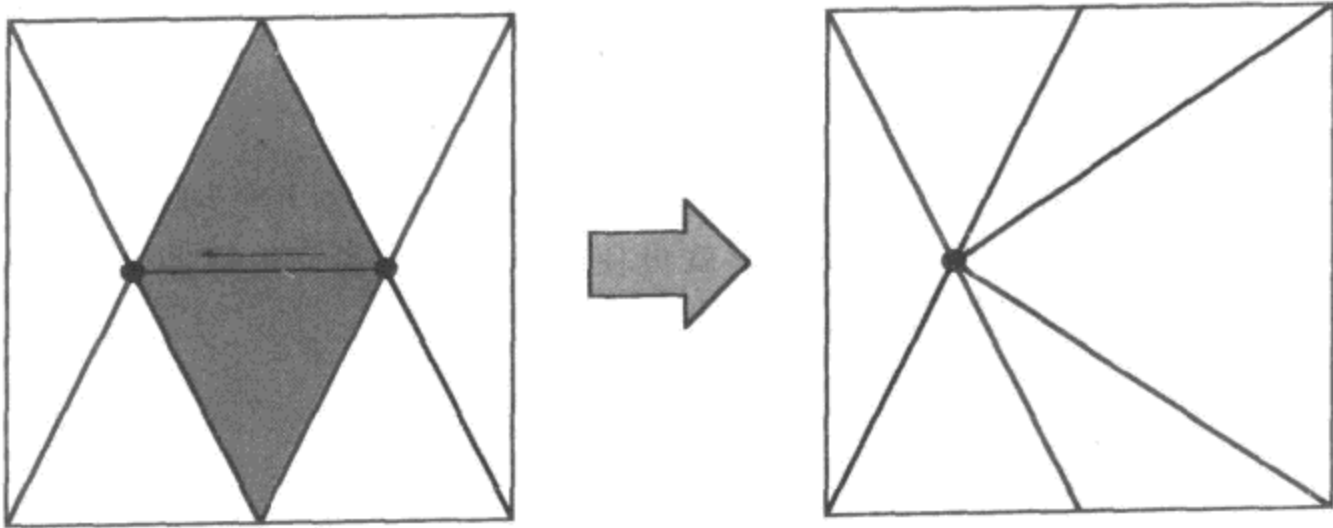


图 9.11 边压缩操作将合并边的两个端点并删除共享该边的两个三角形

通过计算每条边的两个压缩代价(一个端点一个压缩代价),可以确定最终将压缩三角形网格中的哪些边删除。基于三角网格外观的变化量,可以计算出删除一条边中的每个端点的压缩代价。具有最低压缩代价的端点的边将被最先删除。如果一条边被定义为不可删除的边,那么其端点的压缩代价相应地设置为一个较大的值。

有许多计算边的压缩代价的方法。在本节提供的方法中,边的压缩代价由边的长度和三角网格的平坦度共同决定,三角网格的平坦度是指那些环绕将被删除端点的边的两侧三角网格的平坦度。假设要计算图 9.12 中所示的端点 V_1 的删除代价,如图中所示在边压缩

的过程中,端点 V_1 被移到端点 V_2 的位置。首先计算端点 V_1 处的法向量 N , N 为环绕该端点的所有三角形的法向量的平均值(见第6章中6.7.1一节)。定义向量 D 为

$$D = \frac{N \times E}{\|N \times E\|} \quad (9.35)$$

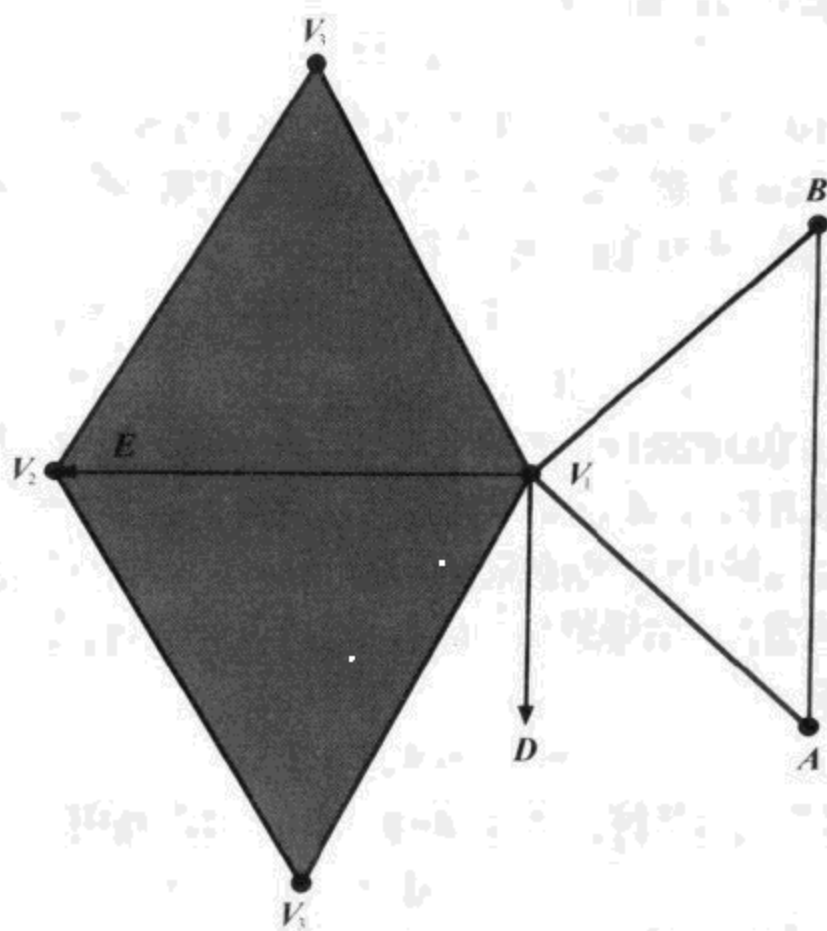


图9.12 计算边压缩的代价

其中 E 为端点 V_1 指向端点 V_2 的向量

$$E = V_2 - V_1 \quad (9.36)$$

向量 D 同时垂直于 V_1 点处表面法线和端点为 V_1 和 V_2 的边。用 D 可以判断一个点位于边的哪一侧。

如果任一端点为 V_1 的边不被两个三角形共享,则端点 V_1 不能删除,否则将会导致三角网格边界形状的改变。如果端点 V_1 确实位于网格内,则对于连接端点 V_1 和 V_2 的边,分别检测与其相邻的两个三角形的第三个端点 V_3 (该端点不位于共享边上)的位置以确定这两个三角形分别位于共享边的正侧还是负侧。如果条件

$$D \cdot (V_3 - V_1) \geq 0 \quad (9.37)$$

成立,那么相应的三角形位于共享边的正侧,否则,位于共享边的负侧。共享一条边的两个三角形必须分别满足上述两个条件,否则,如果两个三角形都在共享边正侧或者都在负侧,那么这条边不能压缩。

令 T_{pos} 代表在公共边正侧的三角形的单位法向量, T_{neg} 代表在同一公共边负侧的三角形的单位法向量。通过比较单位法向量 T_{pos} 和 T_{neg} 与围绕顶点 V_1 的其他三角形的单位法向量,可以估计出在压缩边两侧三角网格的平坦度。对于这些三角形,令 d 为最小点积数

值,这里的点积是指压缩边正侧三角形的法向量与法向量 T_{pos} 之间的点积以及压缩边负侧三角形的法向量与法向量 T_{neg} 之间的点积。如果 d 的值接近 1,则表明在该边两侧三角网格近似于平面,而 d 值越小则说明环绕端点 V_1 的三角形之间存在的夹角越大。如果低于最大表面粗糙程度所对应的某个阈值,那么连接端点 V_1 和 V_2 的边不能被压缩,否则,给该边分配一个压缩代价值 c ,其计算公式如下:

$$c = (1 - d) \|E\| \quad (9.38)$$

为了明确计算最小点积 d 的过程,假设三角形的顶点为 V_1, A 和 B (其中 A 和 B 均不为 V_2),并且其单位法向量为 T 。通过计算式(9.39)所示的点积,将顶点 A 和 B 分成位于压缩边正侧、负侧或者位于压缩边上三种情况。

$$\begin{aligned} a &= D \cdot (A - V_1) \\ b &= D \cdot (B - V_1) \end{aligned} \quad (9.39)$$

a 和 b 分别代表包含压缩边和法向量 D 的平面与三角形顶点 A, B 之间的距离。如果对于某一小距离 $\epsilon, a > \epsilon$ 或者 $b > \epsilon$ 成立,则其对应的顶点位于压缩边的正侧。类似地,如果 $a < -\epsilon$ 或者 $b < -\epsilon$ 成立,则对应的顶点位于压缩边的负侧。与压缩边之间的距离小于等于 ϵ 的点被认为位于压缩边上。如果顶点 A 或者 B 位于边的正侧,且点积 $T \cdot T_{\text{pos}}$ 小于 d 时,则用它替代点积 d ,即

$$d \leftarrow \min\{d, T \cdot T_{\text{pos}}\} \quad (9.40)$$

如果顶点 A 或者 B 位于边的负侧,且点积 $T \cdot T_{\text{neg}}$ 小于 d 时,则用它替代点积 d ,即

$$d \leftarrow \min\{d, T \cdot T_{\text{neg}}\} \quad (9.41)$$

公式(9.40)和公式(9.41)中的操作可能发生在同一个三角形中。

本节提出的边压缩算法允许对图 9.13 所示的边进行压缩。只要在压缩边两侧的三角网格足够平坦,两个方向差别相当大的三角形之间的公共边也可以被压缩掉。

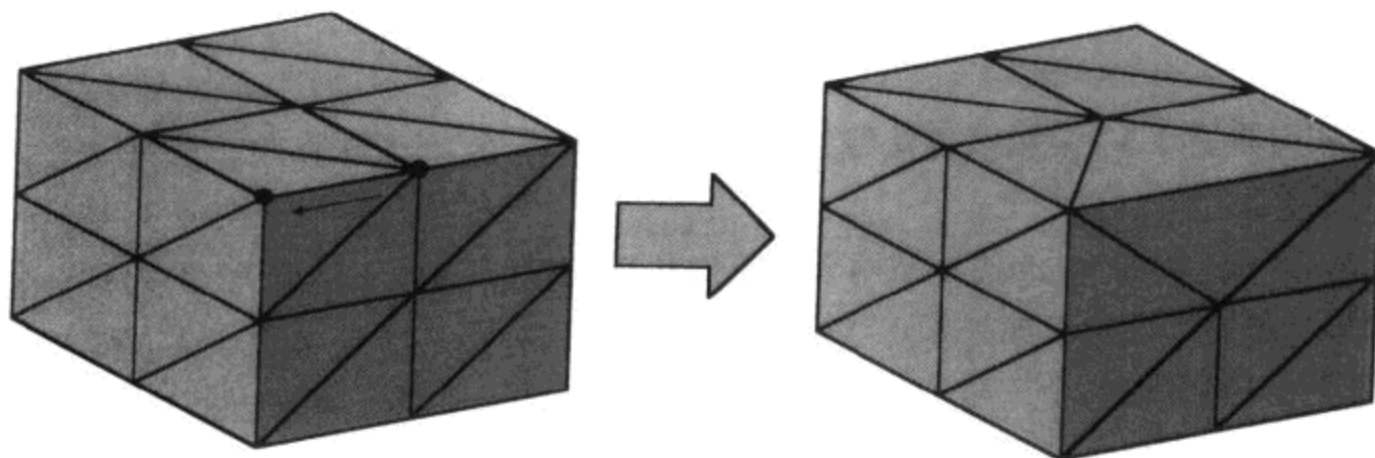


图 9.13 边压缩将两个方向差别相当大的三角形之间的公共边压缩掉

图 9.14 和图 9.15 所示的是人物模型的原始三角网格和用边压缩技术删除了 30% 的三角形后的三角网格。注意三角形高度集中的区域和相对平坦的区域中压缩边是如何首先被删除的。

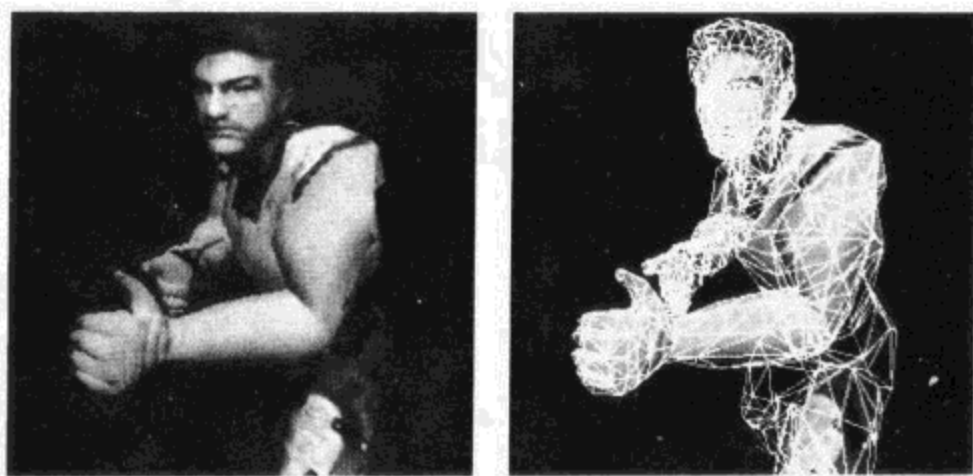


图 9.14 原始人物模型及其三角网格的线框图



图 9.15 删除了 30% 的三角形后同一人物模型及其三角网格的线框图

9.6 本章小结

深度值修正

为了用距离 δ 修正 z 坐标值约为 P_z 的顶点的深度时, 透视投影变换矩阵中的 (3,3) 元素被乘以 $(1+\epsilon)$, 其中

$$\epsilon = -\frac{2fn}{f+n} \left(\frac{\delta}{P_z(P_z + \delta)} \right)$$

贴图

一个中心位于 P 点, 宽为 w , 高为 h 的贴图, 其法线方向为 N , 切线方向为 T , 其裁剪平面为

$$\begin{aligned} \text{left} &= \left(T, \frac{w}{2} - T \cdot P \right) & \text{right} &= \left(-T, -\frac{w}{2} + T \cdot P \right) \\ \text{bottom} &= \left(B, \frac{h}{2} - B \cdot P \right) & \text{top} &= \left(-B, \frac{h}{2} + B \cdot P \right) \\ \text{front} &= (-N, d + N \cdot P) & \text{back} &= (N, d - N \cdot P) \end{aligned}$$

其中 $B = N \times T$, d 为贴图中的所有点与过 P 点的切平面之间的最大距离。贴图中顶点 Q 的纹理坐标为

$$s = \frac{\mathbf{T} \cdot (\mathbf{Q} - \mathbf{P})}{w} + \frac{1}{2}$$

$$t = \frac{\mathbf{B} \cdot (\mathbf{Q} - \mathbf{P})}{h} + \frac{1}{2}$$

公告牌

宽为 w , 高为 h , 中心点为 \mathbf{P} , 方向为 θ 的无约束四边形公告牌的顶点计算方法为

$$\begin{aligned} \mathbf{Q}_1 &= \mathbf{P} + \mathbf{X} + \mathbf{Y} & \mathbf{Q}_2 &= \mathbf{P} - \mathbf{X} + \mathbf{Y} \\ \mathbf{Q}_3 &= \mathbf{P} - \mathbf{X} - \mathbf{Y} & \mathbf{Q}_4 &= \mathbf{P} + \mathbf{X} - \mathbf{Y} \end{aligned}$$

其中

$$\mathbf{X} = \left(\frac{w}{2} \cos \theta \right) \mathbf{R} + \left(\frac{w}{2} \sin \theta \right) \mathbf{U} \quad \mathbf{Y} = \left(-\frac{h}{2} \sin \theta \right) \mathbf{R} + \left(\frac{h}{2} \cos \theta \right) \mathbf{U}$$

并且方向向量 \mathbf{R} 和 \mathbf{U} 分别表示世界坐标系中观察视图的右向和顶向向量。对于被约束得只能绕 z 轴旋转的四边形公告牌, 其顶点的计算公式为

$$\begin{aligned} \mathbf{Q}_1 &= \mathbf{P} + \frac{\mathbf{X}}{\|\mathbf{X}\|} + \left\langle 0, 0, \frac{h}{2} \right\rangle & \mathbf{Q}_2 &= \mathbf{P} - \frac{\mathbf{X}}{\|\mathbf{X}\|} + \left\langle 0, 0, \frac{h}{2} \right\rangle \\ \mathbf{Q}_3 &= \mathbf{P} - \frac{\mathbf{X}}{\|\mathbf{X}\|} - \left\langle 0, 0, \frac{h}{2} \right\rangle & \mathbf{Q}_4 &= \mathbf{P} + \frac{\mathbf{X}}{\|\mathbf{X}\|} - \left\langle 0, 0, \frac{h}{2} \right\rangle \end{aligned}$$

其中

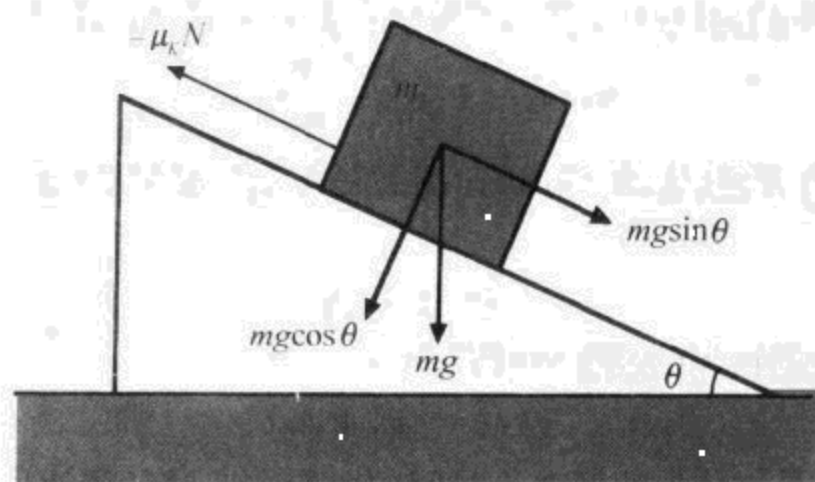
$$\mathbf{X} = \frac{w}{2} \langle -(C_y - P_y), C_x - P_x, 0 \rangle$$

C 为世界坐标系中观察点的位置。

习 题

1. 假设一点到视平截体的近平面的距离 $n=1$, 到远平面的距离 $f=100$, 计算透视投影变换矩阵的(3,3)元素乘以何值才可使中心位于 $z=-20$ 处的模型能够向观察点移动距离 0.2?
2. 如果使用 16 位的 z 缓冲区, 计算使习题 1 中的对象能够向观察点移动距离 d 的最小值。
3. 编写程序实现表面贴图效果。假设贴图的中心为 \mathbf{P} , 法线方向为 \mathbf{N} , 切线方向为 \mathbf{T} , 宽为 w , 高为 h 。程序应能够用贴图的边界平面对任意三角网格进行裁剪, 并且能够计算出贴图中每个顶点的纹理坐标。
4. 实现一个粒子系统, 其中每个粒子作为一个含纹理并且以粒子的位置为中心的正方形进行渲染。每个粒子含有以下参数, 即粒子的空间位置 \mathbf{P} , 粒子的速度 \mathbf{V} , 粒子半径 r (与粒子的高和宽相等) 以及粒子的方向角 θ 。
5. 编程实现在一个光源照射下的三角网格的模板阴影显示。假设三角网格的定义中含有 n 个三角形索引, 每个三角形索引含有顶点数组中的 3 个元素, 顶点数组中含有 m 个顶点元素。该程序应该首先计算一个边表, 然后确定与光源对应的模型轮廓中的边, 并且用程序 9.3 中所示代码中的模板缓冲区操作渲染轮廓边的延伸四边形。

第 10 章 线性物理运动



模拟动态物体的准确动作以及它们之间的相互影响可以增加游戏的真实感,而且这通常不需要特别复杂的数学就可以完成。在这一章和以后的章节,将讨论可以运用于游戏的一些经典力学问题。首先,从线性运动开始,线性运动指除了旋转环境中的运动之外的一切运动。

10.1 位置方程

位置函数是一个物体的三维位置关于时间的函数。当一个物体在初始位置的时间已知时,其他时间就相对该初始位置的时间进行计算。例如,一个物体以速度 v_0 作直线运动,如果在时间 $t=0$ 时的位置为 x_0 ,则其后任意时刻的位置 $x(t)$ 都可以用以下函数表示,即

$$x(t) = x_0 + v_0 t \quad (10.1)$$

物体的速度函数是物体的三维速度关于时间的函数,速度函数 $v(t)$ 是位置函数对时间 t 的一阶导数,时间导数通常以一个点标注在求导方程上以示区别,即

$$v(t) = \dot{x}(t) = \frac{d}{dt}x(t) \quad (10.2)$$

既然物体位置方程已由方程(10.1)给出,速度又是不变的常数,那么速度函数 $v(t)$ 就可简化为

$$v(t) = v_0 \quad (10.3)$$

以恒定加速度 a_0 运动的物体的速度方程则为

$$v(t) = v_0 + a_0 t \quad (10.4)$$

物体加速度函数 $a(t)$ 是物体加速度关于时间的函数。通过计算速度函数关于时间的一阶导数可得加速度函数为

$$a(t) = \dot{v}(t) = \ddot{x}_0(t) = \frac{d^2}{dt^2}x(t) \quad (10.5)$$

可以对任意速度函数求积分,从而确定一个运动的物体从时刻 t_1 运动到 t_2 时刻的路程为

$$d = \int_{t_1}^{t_2} v(t) dt \quad (10.6)$$

计算方程(10.4)从时间 0 到时间 t 的积分,可以得到

$$d = \int_0^t (v_0 + a_0 t) dt = v_0 t + \frac{1}{2} a_0 t^2 \quad (10.7)$$

加上初始位置 x_0 ,则得一个以均匀加速度运动的物体的位置函数

$$x(t) = x_0 + v_0 t + \frac{1}{2} a_0 t^2 \quad (10.8)$$

通常,我们感兴趣的是物体在外力作用下的运动,而且希望找到描述这种运动中物体位置的函数。假设作用于物体的外力为 F_1, F_2, \dots, F_N ,它们之和等于该物体的质量与其加速度 $a(t)$ 之积,即

$$\sum_{i=1}^N F_i(t) = ma(t) = m\ddot{x}(t) \quad (10.9)$$

每一个作用在物体上的外力可能是一个不变的常量,也可能是关于物体位置的函数,还可能是关于物体速度的函数。方程(10.9)是一个二阶微分方程,它的解 $x(t)$ 是物体的位置函数。在10.2节将回顾二阶微分方程的一般解。将在本章及以后章节中相应的地方介绍一些具体的外力方程的解。

10.2 二阶微分方程

方程 $x(t)$ 的二阶线性微分方程的一般形式如下:

$$\frac{d^2}{dt^2}x(t) + a \frac{d}{dt}x(t) + bx(t) = f(t) \quad (10.10)$$

用基本微分符号表示导数,上式可表示成更简洁的形式为

$$x''(t) + ax'(t) + bx(t) = f(t) \quad (10.11)$$

本章中字母 a 和 b 总是常数,但在一般情况下它们可能是 t 的函数。

10.2.1 齐次方程

函数 $f(t)$ 等于 0 时的微分方程称为齐次方程,如

$$x''(t) + ax'(t) + bx(t) = 0 \quad (10.12)$$

在寻找方程(10.12)的解 $x(t)$ 之前,先做两个重要的分析。首先,假设函数 $x_1(t)$ 和 $x_2(t)$ 分别是方程(10.12)的两个解,那么函数 $Ax_1(t)$ 和 $Bx_2(t)$ 也是该方程的解,其中 A 与 B 是任意常数,进而函数 $Ax_1(t) + Bx_2(t)$ 也是该方程的解,这是因为

$$\begin{aligned} & Ax_1''(t) + Bx_2''(t) + a[Ax_1'(t) + Bx_2'(t)] + b[Ax_1(t) + Bx_2(t)] \\ &= A[x_1''(t) + ax_1'(t) + bx_1(t)] + B[x_2''(t) + ax_2'(t) + bx_2(t)] \\ &= A \cdot 0 + B \cdot 0 = 0 \end{aligned} \quad (10.13)$$

方程(10.12)的一般解 $x(t)$ 在进行置换时也非常容易,即令

$$x(t) = e^{rt} \quad (10.14)$$

$x(t)$ 的一阶和二阶导数为

$$\begin{aligned} x'(t) &= re^{rt} \\ x''(t) &= r^2e^{rt} \end{aligned} \quad (10.15)$$

把它们都代入方程(10.12)中得

$$r^2e^{rt} + are^{rt} + be^{rt} = 0 \quad (10.16)$$

再给方程两边同时乘以 e^{-rt} , 消去指数项得

$$r^2 + ar + b = 0 \quad (10.17)$$

方程(10.17)叫做方程(10.12)的辅助方程,而且它的解为

$$\begin{aligned} r_1 &= -\frac{a}{2} + \frac{1}{2}\sqrt{a^2 - 4b} \\ r_2 &= -\frac{a}{2} - \frac{1}{2}\sqrt{a^2 - 4b} \end{aligned} \quad (10.18)$$

除非 $r_1 = r_2$, 方程(10.12)的一般解为

$$x(t) = Ae^{r_1 t} + Be^{r_2 t} \quad (10.19)$$

【例 10.1】 解微分方程

$$x''(t) + 5x'(t) + 6x(t) = 0 \quad (10.20)$$

解 方程(10.20)辅助方程为

$$r^2 + 5r + 6 = 0 \quad (10.21)$$

它的解为 $r_1=2$ 和 $r_2=3$,那么方程(10.20)的一般解就如前面所给出的那样,为

$$x(t) = Ae^{2t} + Be^{3t} \quad (10.22)$$

其中 A 和 B 都是任意常数。

如果 $r_1=r_2$,那么一定有 $a^2=4b$,这样方程(10.12)可以写为

$$x''(t) + ax'(t) + \frac{a^2}{4}x(t) = 0 \quad (10.23)$$

容易证明函数

$$x(t) = te^{-(a/2)t} \quad (10.24)$$

为方程(10.23)的一个解,于是当 $r_1=r_2$ 时,方程(10.12)的一般解就是

$$x(t) = Ae^{rt} + Bte^{rt} \quad (10.25)$$

其中, $r_1=r_2=r$ 。

如果 $a^2-4b<0$,那么辅助方程的根就是复数。方程(10.19)的解仍然正确,但它要求用到复数计算。用以下公式

$$e^{a+\beta i} = e^a(\cos\beta + i\sin\beta) \quad (10.26)$$

可以以实值函数形式完全表达方程的解,参考附录 A.4 中的介绍。假设 a, b 都是实数,辅助方程的二根 r_1, r_2 为共轭的复数根,那么 r_1, r_2 可以写为

$$\begin{aligned} r_1 &= \alpha + \beta i \\ r_2 &= \alpha - \beta i \end{aligned} \quad (10.27)$$

其中

$$\begin{aligned} \alpha &= -\frac{a}{2} \\ \beta &= \frac{1}{2}\sqrt{4b-a^2} \end{aligned} \quad (10.28)$$

这样,方程(10.19)的解可以写为

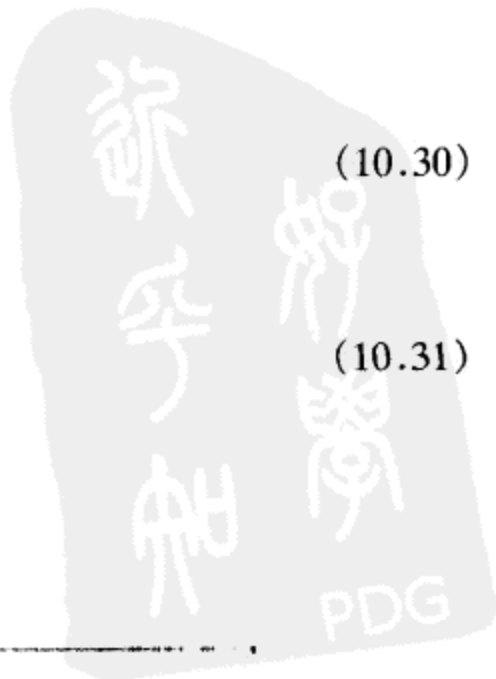
$$\begin{aligned} x(t) &= Ae^{(\alpha+\beta i)t} + Be^{(\alpha-\beta i)t} \\ &= Ae^{\alpha t}(\cos\beta t + i\sin\beta t) + Be^{\alpha t}(\cos\beta t - i\sin\beta t) \\ &= e^{\alpha t}[(A+B)\cos\beta t + (A-B)i\sin\beta t] \end{aligned} \quad (10.29)$$

方程的解可以用两个实数 C_1 和 C_2 表示,如令

$$\begin{aligned} A &= \frac{1}{2}(C_1 + C_2 i) \\ B &= \frac{1}{2}(C_1 - C_2 i) \end{aligned} \quad (10.30)$$

将这些值代入方程(10.29)可得

$$x(t) = e^{\alpha t}(C_1\cos\beta t + C_2\sin\beta t) \quad (10.31)$$

【例 10.2】 解微分方程

$$x''(t) + 4x(t) = 0 \quad (10.32)$$

解 微分方程的辅助方程为

$$r^2 + 4 = 0 \quad (10.33)$$

其解为 $r_1 = 2i$ 和 $r_2 = -2i$, 方程(10.32)的解可以写成

$$x(t) = Ae^{2it} + Be^{-2it} \quad (10.34)$$

但也可以用方程(10.31)给出的完整的实值形式函数表示的方程(10.32)的解为

$$x(t) = C_1 \cos 2t + C_2 \sin 2t \quad (10.35)$$

其中 C_1, C_2 都是任意常数。

通过引入常数 $D = \sqrt{C_1^2 + C_2^2}$, 方程(10.31)可以转换为仅包括一个单一的三角函数的解, 即

$$x(t) = De^{i\beta t} \left(\frac{C_1}{D} \cos \beta t + \frac{C_2}{D} \sin \beta t \right) \quad (10.36)$$

假设 C_1, C_2 分别为一个直角三角形的两条直角边的边长, 而且角 δ 是 C_1 边的对角(见图 10.1)。则 D 即为直角三角形的斜边的长度, 于是

$$\begin{aligned} \frac{C_1}{D} &= \sin \delta \\ \frac{C_2}{D} &= \cos \delta \end{aligned} \quad (10.37)$$

把它们代入方程(10.36)得

$$x(t) = D \cos \beta t \sin \delta + D \sin \beta t \cos \delta \quad (10.38)$$

使用和角恒等式(B.13)(见附录 B.4 部分), 上式可表示成

$$x(t) = D \sin(\beta t + \delta) \quad (10.39)$$

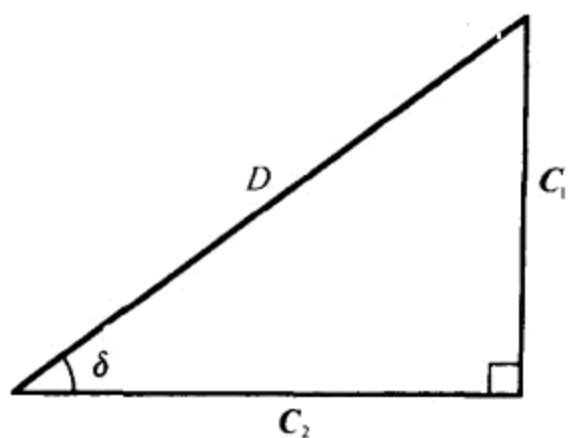


图 10.1 直角三角形

10.2.2 非齐次方程

微分方程

$$x''(t) + ax'(t) + bx(t) = f(t) \quad (10.40)$$

如果 $f(t)$ 不等于 0, 该方程为非齐次方程, 它的解的形式为

$$x(t) = g(t) + p(t) \quad (10.41)$$

其中函数 $g(t)$ 为以下齐次方程的解, 即

$$x''(t) + ax'(t) + bx(t) = 0 \quad (10.42)$$

而函数 $p(t)$ 叫做非齐次方程的特解, 且满足以下方程

$$p''(t) + ap'(t) + bp(t) = f(t) \quad (10.43)$$

为了验证 $g(t) + p(t)$ 确实为方程(10.40)的解, 将上式代入方程得

$$\begin{aligned} & g''(t) + p''(t) + a[g'(t) + p'(t)] + b[g(t) + p(t)] \\ &= g''(t) + ag'(t) + bg(t) + p''(t) + ap'(t) + bp(t) \\ &= 0 + f(t) = f(t) \end{aligned} \quad (10.44)$$

求解非齐次微分方程特解的方法有很多。本节介绍的方法叫“待定系数法”, 它可以满足本章中的所有非齐次方程求解运算。待定系数法的大致思路就是借助已经掌握的有关函数 $f(t)$ 的知识猜测特解 $p(t)$ 的形式。通常将函数选择为与函数 $f(t)$ 有着同样形式或导数与函数 $f(t)$ 有同样形式的项的和, 这个方法很有效。每个项都乘以一个未知系数, 然后可以试着将 $p(t)$ 代入非齐次方程来最终确定它, 如果 $p(t)$ 满足方程(10.40), 则系数可被确定, 从而找到原非齐次方程的特解。下面的例子详细地说明了这个方法。

【例 10.3】 解微分方程

$$x''(t) + 5x'(t) + 6x(t) = 12t - 4 \quad (10.45)$$

解 已知例 10.1 中所示的齐次方程的一般解 $g(t)$ 为

$$g(t) = Ae^{2t} + Be^{3t} \quad (10.46)$$

方程(10.45)的非齐次部分是一个线性多项式, 假设特解的形式为

$$p(t) = Dt^2 + Et + F \quad (10.47)$$

其中, 系数 D, E 和 F 为待定系数, 将 $p(t)$ 代入方程(10.45)中得

$$\begin{aligned} 12t - 4 &= 2D + 5(2Dt + E) + 6(Dt^2 + Et + F) \\ &= 6Dt^2 + (10D + 6E)t + 2D + 5E + 6F \end{aligned} \quad (10.48)$$

令方程两边关于 t 的同类项的系数相等, 可得

$$D = 0, \quad E = 2, \quad F = -1 \quad (10.49)$$

结果, 函数 $p(t) = 2t - 1$ 就是方程(10.45)的一个特解。方程的全解则为

$$\begin{aligned} x(t) &= g(t) + p(t) \\ &= Ae^{2t} + Be^{3t} + 2t - 1 \end{aligned} \quad (10.50)$$

其中, A 和 B 是任意常数。

【例 10.4】 解微分方程

$$x''(t) + 4x(t) = 12\sin t \quad (10.51)$$

解 在例 10.2 中已知齐次方程的一般解 $g(t)$ 为

$$g(t) = A\cos 2t + B\sin 2t \quad (10.52)$$

可以把 $g(t)$ 等价地写成如下形式, 即

$$g(t) = C\sin(2t + \delta) \quad (10.53)$$

由于方程(10.51)的非齐次部分是一个正弦函数, 故假设特解为

$$p(t) = D\sin t + E\cos t \quad (10.54)$$

其中 D 和 E 为待定系数, 把 $p(t)$ 代入方程(10.51)中得

$$12\sin t = -D\sin t - E\cos t + 4(D\sin t + E\cos t) = 3D\sin t + 3E\cos t \quad (10.55)$$

令等式两边正弦函数与余弦函数项的对应系数相等,得

$$D = 4, \quad E = 0 \quad (10.56)$$

结果,函数 $p(t) = 4\sin t$ 就是方程(10.51)的一个特解,全解则为

$$x(t) = A\cos 2t + B\sin 2t + 4\sin t \quad (10.57)$$

或者表示为

$$x(t) = C\sin(2t + \delta) + 4\sin t \quad (10.58)$$

其中系数 A, B 和 C 都是任意常数。

10.2.3 初始条件

迄今为止,所有二阶微分方程的解都含有两个任意常数,利用这两个常数,可以为实际情况定义某个特殊的初始条件,这个初始条件决定了 $t=0$ 时 $x(t)$ 与 $x'(t)$ 的值。假设 $x(t)$ 的最初值是 x_0 , 且 $x'(t)$ 的初始值是 v_0 , 那么函数 $x(t)$ 中的两个任意常量就可由方程(10.59)决定,即

$$\begin{aligned} x(0) &= x_0 \\ x'(0) &= v_0 \end{aligned} \quad (10.59)$$

下面例子说明了上述过程。

【例 10.5】 解微分方程

$$x''(t) + 5x'(t) + 6x(t) = 0 \quad (10.60)$$

其初始条件为

$$\begin{aligned} x(0) &= 3 \\ x'(0) &= 0 \end{aligned} \quad (10.61)$$

解 用例 10.1 给出的微分方程的一般解,即

$$x(t) = Ae^{2t} + Be^{3t} \quad (10.62)$$

$x(t)$ 的一阶导数为

$$x'(t) = 2Ae^{2t} + 3Be^{3t} \quad (10.63)$$

引入式(10.61)给出的初始条件,有

$$\begin{aligned} x(0) &= A + B = 3 \\ x'(0) &= 2A + 3B = 0 \end{aligned} \quad (10.64)$$

解这个线性方程组得

$$A = 9, B = -6 \quad (10.65)$$

这样,满足初始条件的微分方程的解是

$$x(t) = 9e^{2t} - 6e^{3t} \quad (10.66)$$

【例 10.6】 解微分方程

$$x''(t) + 4x(t) = 12\sin t \quad (10.67)$$

其初始条件为

$$\begin{aligned} x(0) &= 0 \\ x'(0) &= 6 \end{aligned} \quad (10.68)$$

解 利用例 10.4 中给出的微分方程的一般解,即

$$x(t) = A\cos 2t + B\sin 2t + 4\sin t \quad (10.69)$$

函数 $x(t)$ 的导数为

$$x'(t) = -2A\sin 2t + 2B\cos 2t + 4\cos t \quad (10.70)$$

将式(10.68)中的初始条件代入得

$$\begin{aligned} x(0) &= A = 0 \\ x'(0) &= 2B + 4 = 6 \end{aligned} \quad (10.71)$$

即得 $B=1$ 。这样,满足初始条件的微分方程的解为

$$x(t) = \sin 2t + 4\sin t \quad (10.72)$$

10.3 抛物运动

这一节,将研究只在重力作用下物体的运动。本章约定,世界坐标系中的 z 坐标轴竖直向上,那么重力加速度就是竖直向下的向量 \mathbf{g} 。

$$\mathbf{g} = \langle 0, 0, -g \rangle \quad (10.73)$$

其中,在地球表面标量 g 大约等于 9.8m/s^2 ,一个处于重力场中的质量为 m 的物体受到的重力为 mg 。

一个抛物体在时间 $t=0$,初始位移为 \mathbf{x}_0 ,初速度为 \mathbf{v}_0 的位置方程 $\mathbf{x}(t)$ 为

$$\mathbf{x}(t) = \mathbf{x}_0 + \mathbf{v}_0 t + \frac{1}{2} \mathbf{g} t^2 \quad (10.74)$$

由于 \mathbf{g} 的 x 和 y 分量全为 0,所以方程(10.74)中只有 z 分量为二次项。位置方程 $\mathbf{x}(t)$ 的三个分量 $x(t)$, $y(t)$ 和 $z(t)$ 分别为

$$\begin{aligned} x(t) &= x_0 + v_x t \\ y(t) &= y_0 + v_y t \\ z(t) &= z_0 + v_z t - \frac{1}{2} g t^2 \end{aligned} \quad (10.75)$$

其中, x_0, y_0 和 z_0 是最初位置的三个分量, v_x, v_y 和 v_z 是初速度的三个分量。当一个被抛出的物体达到其最大高度时,它的垂直速度为 0,通过解方程(10.76)可以得出垂直速度为 0 时的时间 t ,即解

$$\dot{z}(t) = v_z - gt = 0 \quad (10.76)$$

这样,抛物体达到其最大高度的时间为

$$t = \frac{v_z}{g} \quad (10.77)$$

代入方程 $z(t)$,得抛物体能达到的最大高度 h 为

$$h = z_0 + \frac{v_z^2}{2g} \quad (10.78)$$

【例 10.7】 在一个高度为 10m 的平台上,将物体以 50m/s 的初速度抛射出去,速度和水平平面的夹角为 70° ,如图 10.2 所示。求物体可以达到的离地面的最大高度是多少?

解 抛物体的初始高度 z_0 ,向上的初始速度 v_z 为

$$\begin{aligned} z_0 &= 10\text{m} \\ v_z &= 50\sin 70^\circ \approx 47.0\text{m/s} \end{aligned} \quad (10.79)$$

把这些值代入方程(10.78),并且已知 g 的值为 9.8m/s^2 ,这样可得 $h \approx 123\text{m}$ 。

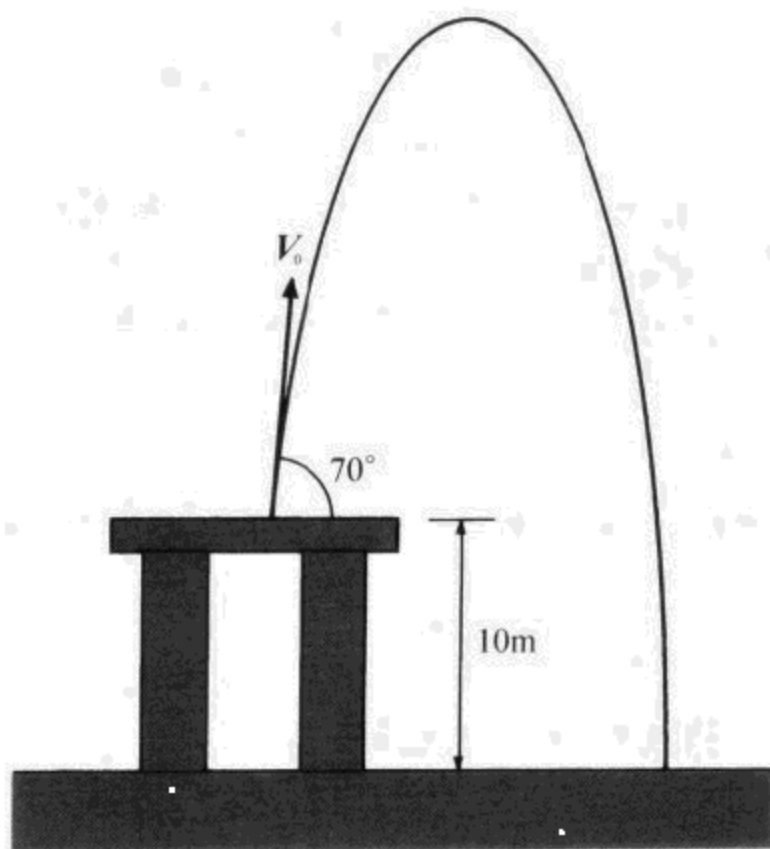


图 10.2 抛物体的路径

一个抛物体在它落回被抛射的平面之前所经过的一段水平距离叫做它的“射程”,如果一个物体从水平面被抛出时初始位置 $z_0 = 0$,那它落回被抛出的水平面的时间 t 可通过解以下方程得到:

$$v_z t - \frac{1}{2} g t^2 = 0 \quad (10.80)$$

方程的一个解为 $t = 0$,它表示物体被抛起时刻的时间。方程的另一个解为

$$t = \frac{2v_z}{g} \quad (10.81)$$

它恰好是物体达到最大高度所需时间的二倍,假定抛物体的运动轨迹位于 xz 平面内,将这个时间代入函数 $x(t)$,并减去 x 坐标的初始值 x_0 ,则得到抛物体的射程 r 为

$$r = \frac{2v_x v_y}{g} \quad (10.82)$$

【例 10.8】 将一个物体以 30m/s 的初速度投射出去,初速度与地面的夹角为 40° ,如图 10.3 所示。求该物体在落地前的飞行距离。

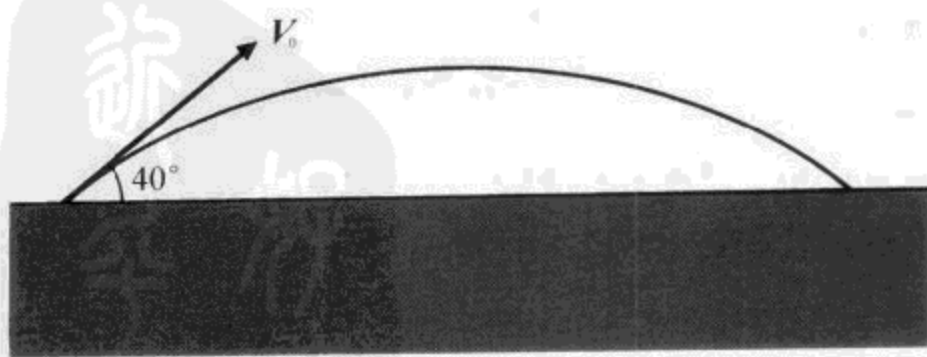


图 10.3 抛物体的路径

解 假设将物体从原点抛出且其路径位于 xz 平面, 则初速度的 x 和 z 分量 v_x 和 v_z 分别是

$$\begin{aligned} v_x &= 30\cos 40^\circ \approx 23.0\text{m/s} \\ v_z &= 30\sin 40^\circ \approx 19.3\text{m/s} \end{aligned} \quad (10.83)$$

将它们代入方程(10.82)中, 且令 g 为 9.8m/s^2 , 那么抛物体的射程为 90.4m 。

给定物体的最初速度 s , 通过调整速度的方向, 可以使抛物体达到一定的高度或者达到一定射程。对于 xz 平面中的抛物运动, 抛物体初速度的 x 和 z 分量 v_x 和 v_z 分别是

$$\begin{aligned} v_x &= s\cos\alpha \\ v_z &= s\sin\alpha \end{aligned} \quad (10.84)$$

α 是抛物体的初始角度与水平面的夹角, 给出一个所要达到的最大高度值 h , 将 v_z 带入方程(10.78)中, 解 α 得

$$\alpha = \sin^{-1}\left(\frac{1}{s}\sqrt{2g(h-z_0)}\right) \quad (10.85)$$

给出射程 r , 将 v_x 和 v_z 的值代入方程(10.82)得

$$r = \frac{2s^2}{g}\sin\alpha\cos\alpha = \frac{s^2}{g}\sin 2\alpha \quad (10.86)$$

对 α 进行求解得

$$\alpha = \frac{1}{2}\sin^{-1}\frac{rg}{s^2} \quad (10.87)$$

由于 $\sin(\pi - \alpha) = \sin\alpha$, 所以解方程(10.86)可得产生同一射程 r 的两个角度, 即方程(10.87)给出的角度 α 及其余角 $\pi/2 - \alpha$ 。如果方程(10.85)和方程(10.87)中的反正弦函数的参数为大于 1 的值, 则说明抛物体的初速度 s 不够大, 没办法达到最大高度或者射程。

【例 10.9】 如图 10.4 所示, 一个物体从地面以 65m/s 的初速度发射出去, 假设地面平坦, 求以什么样的发射角度 α 发射才可以让它的射程达到 400m 。

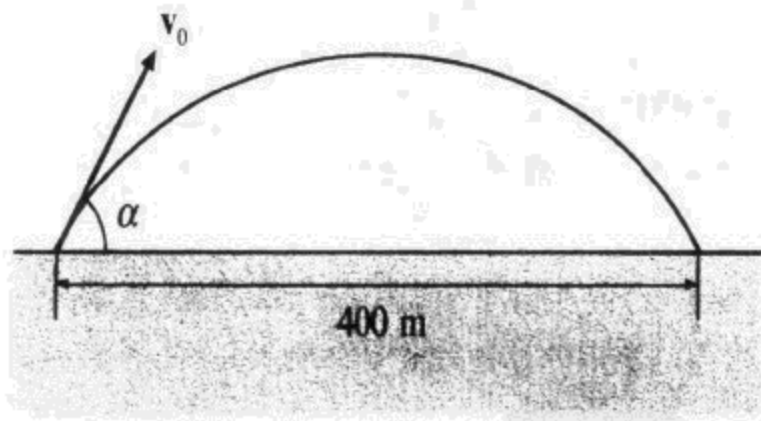


图 10.4 抛物体的发射角

解 将速度值 $v = 65\text{m/s}$ 以及 $r = 400\text{m}$ 带入方程(10.87)中去, 可以得到角度 $\alpha \approx 34^\circ$, 它的余角 $\beta = 56^\circ$, 也同样使物体射程达到 400m , 如果采用角度 α , 则初始速度 v_x 和 v_z 分别是

$$\begin{aligned} v_x &= 65\cos 34^\circ \approx 53.9\text{m/s} \\ v_z &= 65\sin 34^\circ \approx 36.3\text{m/s} \end{aligned} \quad (10.88)$$

10.4 阻尼运动

在前面研究物体运动时总将其所受阻力忽略不计,实际上,一个物体在运动时它的速度总会越来越慢,这就是受阻力的作用,这些阻力也许来自空气、水或其他物质。受阻力的物体的精确运动方程是复杂的,但可以近似地认为物体所受阻力的方向为物体运动的反方向,而且所受阻力大小也与物体速度的大小成正比例。

一个质量为 m 的物体在重力和来自环境阻力的作用下的受力方程为

$$mg - mk\dot{x}(t) = m\ddot{x}(t) \quad (10.89)$$

其中, mk 是常量,表示阻力的大小。上式可以写成一个二阶非齐次微分方程,即

$$\ddot{x} + k\dot{x}(t) = g \quad (10.90)$$

利用待定系数法可得方程(10.90)的一个特解为

$$x(t) = \frac{g}{k}t \quad (10.91)$$

再加上齐次微分方程的一般解,可得

$$x(t) = A + Be^{-kt} + \frac{g}{k}t \quad (10.92)$$

其中,向量 A 与 B 都是任意常量,它们可以由初始条件决定。给定初始位移 x_0 和初速度 v_0 ,有

$$\begin{aligned} x(0) &= x_0 \\ \dot{x}(0) &= v_0 \end{aligned} \quad (10.93)$$

它们分别等于 $t=0$ 时函数 $x(t)$ 和 $\dot{x}(t)$ 值,则得以下方程组

$$\begin{aligned} A + B &= x_0 \\ -kB + \frac{g}{k} &= v_0 \end{aligned} \quad (10.94)$$

解该方程组可得向量 A 与 B 的表达式

$$\begin{aligned} A &= x_0 - \frac{g}{k^2} + \frac{v_0}{k} \\ B &= \frac{g}{k^2} - \frac{v_0}{k} \end{aligned} \quad (10.95)$$

在物体作阻尼运动时的位置函数 $x(t)$ 为

$$x(t) = x_0 + \frac{g}{k}t + \frac{kv_0 - g}{k^2}(1 - e^{-kt}) \quad (10.96)$$

对 $x(t)$ 求导就可以得到速度函数 $v(t)$,其表达式为

$$v(t) = \dot{x}(t) = \frac{g}{k} + \left(v_0 - \frac{g}{k}e^{-kt}\right) \quad (10.97)$$

随着时间推移,作阻尼运动物体的运动速度最终会趋于一个常量,称其为末端速度,当 t 趋于无穷大时,速度 $v(t)$ 的极限值即为末端速度 v_T ,即

$$v_T = \lim_{t \rightarrow \infty} v(t) = \frac{g}{k} \quad (10.98)$$

方程(10.96)所示的位置函数的物理意义不十分明显,但是在 k 趋于 0 时,该位置函数就会更接近位置方程(10.74)。通过计算以下极限可以求出:

$$x(t) = \lim_{k \rightarrow 0} \left[x_0 + \frac{g}{t}t + \frac{kv_0 - g}{k^2}(1 - e^{-kt}) \right] \quad (10.99)$$

用幂级数代替指数函数(见附录 D 中的公式(D.11)),可得

$$\begin{aligned} x(t) &= \lim_{k \rightarrow 0} \left[x_0 + \frac{g}{t}t + \frac{kv_0 - g}{k^2} \left(kt - \frac{k^2 t^2}{2!} + \frac{k^3 t^3}{3!} - \frac{k^4 t^4}{4!} + \dots \right) \right] \\ &= \lim_{k \rightarrow 0} \left[x_0 + \frac{g}{t}t + (kv_0 - g) \left(\frac{t}{k} - \frac{t^2}{2!} + \frac{kt^3}{3!} - \frac{k^2 t^4}{4!} + \dots \right) \right] \\ &= \lim_{k \rightarrow 0} \left[x_0 + v_0 t - \frac{1}{2} kv_0 t^2 + \frac{1}{2} g t^2 + (kv_0 - g) \left(\frac{kt^3}{3!} - \frac{k^2 t^4}{4!} + \dots \right) \right] \\ &= x_0 + v_0 t + \frac{1}{2} g t^2 \end{aligned} \quad (10.100)$$

10.5 摩擦力

摩擦力是一种常见的力,当两个物体表面接触时就会产生摩擦力。本节仅讨论两种摩擦力,即滑动摩擦力和静摩擦力。滑动摩擦力产生在两个相对运动的物体之间,静摩擦力是指一个静止物体放在另一个物体表面上时产生的力。

当一个物体在另一物体表面滑动时,所受到的阻力是非常复杂的,但用式(10.101)可以非常近似地表示滑动摩擦力,即

$$F_k = -\mu_K N \quad (10.101)$$

其中, N 是正压力(大多为重力),该力把物体约束在另一物体表面。 μ_K 叫做滑动摩擦系数,式(10.101)中的负号表示摩擦力的方向与物体运动方向相反,摩擦系数 μ_K 是一个常量,它取决于两个物体的接触表面的材料。表 10.1 列出了典型材料之间的滑动摩擦系数 μ_K 和静摩擦系数 μ_S 。

表 10.1 典型材料之间的滑动摩擦系数和静摩擦系数

表面	μ_K	μ_S
铝和铝	1.4	1.1
铝和钢	0.47	0.61
铜和钢	0.36	0.53
铜和铜	0.57	0.74
镍和镍	0.53	1.1
玻璃和玻璃	0.40	0.94
铜和玻璃	0.53	0.68
橡木和橡木(纹理平行)	0.48	0.62
橡木和橡木(纹理垂直)	0.32	0.54

续表

表面	μ_K	μ_S
橡胶和水泥(干)	0.90	1.0
橡胶和水泥(湿)	0.25	0.30

【例 10.10】 假设一个 10kg 的物体从一个倾角 30° 的斜面滑下,若 $\mu_K = 0.5$,求此物体的加速度。

解 设物体的质量为 m ,平面倾角为 θ ,如图 10.5 所示。物体受重力和摩擦阻力,重力可以分解为垂直于斜面和垂直于斜面的两个分力,其中平行于斜面的分力 F_G 为

$$F_G = mg \sin \theta \quad (10.102)$$

该力使物体沿斜面下滑。重力与斜面垂直的分力 N 为

$$N = mg \cos \theta \quad (10.103)$$

该力阻止物体沿斜面下滑。滑动摩擦力 F_K 为

$$F_K = -\mu_K N = \mu_K mg \cos \theta \quad (10.104)$$

其方向与 F_G 相反,物体的加速度等于作用于物体的合力除以其质量,即

$$a = \frac{F_G + F_K}{m} = g \sin \theta - \mu_K g \cos \theta \quad (10.105)$$

代入斜面的角度与摩擦系数,可以得

$$a = (9.8 \text{m/s}^2) \cdot \frac{1}{2} - 0.5 \cdot (9.8 \text{m/s}^2) \cdot \frac{\sqrt{3}}{2} \approx 0.656 \text{m/s}^2 \quad (10.106)$$

可以看出加速度与物体的质量无关。

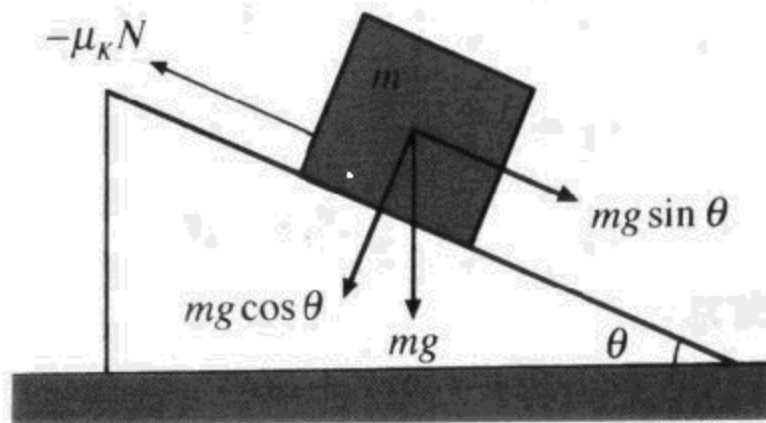


图 10.5 例 10.10 中用到的物体

静摩擦力总是阻止一个运动物体在另一个物体表面上的运动,该力与作用于物体上的切向力相反。由静摩擦产生的静摩擦力 F_S 的最大值为

$$F_S = -\mu_S N \quad (10.107)$$

其中, N 是一个正压力, μ_S 为静摩擦系数,负号表示该力的方向总是与企图使物体产生运动的力的方向相反。 μ_S 在表 10.1 中可以查出。

一旦作用于物体的力超过方程(10.107)给出的最大静摩擦力 F_S ,则物体开始运动,并且静摩擦力被滑动摩擦力 F_K 取代,显然 $F_K < F_S$,所以一旦一个物体开始运动,一个较小的

力就可以保持物体的运动,也就是说需要一个较大的力使物体开始运动。

【例 10.11】 一个物体在一个平面上静止,它们之间的静摩擦系数为 μ_s ,求平面倾角为多少度时物体在重力作用下开始下滑。

解 需要确定什么时候重力平行于平面的分量超过静摩擦力,即当下述条件成立时物体开始运动:

$$mg\sin\theta = \mu_s N = \mu_s mg\cos\theta \quad (10.108)$$

其中, θ 为平面倾角。解 θ 得

$$\theta = \tan^{-1}\mu_s \approx 26.6^\circ \quad (10.109)$$

10.6 本章小结

受力方程

物体的加速度 $a(t)$ 乘以其质量 m 等于作用于物体的外力之和:

$$\sum_{i=1}^N \mathbf{F}_i(t) = m\mathbf{a}(t) = m\ddot{\mathbf{x}}(t)$$

二阶微分方程

齐次二阶微分方程的一般形式为

$$x''(t) + ax'(t) + bx(t) = 0$$

其一般解为

$$x(t) = Ae^{r_1 t} + Be^{r_2 t}$$

其中

$$r_1 = -\frac{a}{2} + \frac{1}{2}\sqrt{a^2 - 4b}$$

$$r_2 = -\frac{a}{2} - \frac{1}{2}\sqrt{a^2 - 4b}$$

如果 $r_1 = r_2 = r$, 则一般解为

$$x(t) = Ae^{rt} + Bte^{rt}$$

如果 r_1 和 r_2 是复数, 则方程的通解也可以写为

$$x(t) = e^{\alpha t}(C_1 \cos\beta t + C_2 \sin\beta t)$$

其中

$$\alpha = -\frac{a}{2}$$

$$\beta = \frac{1}{2}\sqrt{4b - a^2}$$

可以等效地表示为

$$x(t) = D\sin(\beta t + \delta) \quad (10.39)$$

其中

$$D = \sqrt{C_1^2 + C_2^2}$$

$$\delta = \sin^{-1} \frac{C_1}{D}$$

抛物运动

抛物运动的位置方程 $x(t)$ 为

$$x(t) = x_0 + v_0 t + \frac{1}{2} g t^2$$

其中 x_0 为初始位置, v_0 为初速度, $g = \langle 0, 0, -g \rangle$ 是重力加速度。抛物体可达到的最高位置为

$$h = z_0 + \frac{v_z^2}{2g}$$

抛物体的射程 r 为

$$r = \frac{2v_x v_y}{g}$$

阻尼运动

作受阻运动的物体的位置方程 $x(t)$ 为

$$x(t) = x_0 + \frac{g}{k} t + \frac{k v_0 - g}{k^2} (1 - e^{-kt}),$$

其中, k 代表阻尼力的饱和度。末端速度 v_T 为

$$v_T = \frac{g}{k}$$

摩擦力

滑动摩擦力 F_K 为

$$F_K = -\mu_K N$$

其中 μ_K 为滑动摩擦系数。滑动摩擦力的方向与物体运动的方向相反。

最大静摩擦力为

$$F_S = -\mu_S N$$

这里 μ_S 为静摩擦系数。静摩擦力的方向与物体运动趋势的方向相反。

习 题

1. 解微分方程

$$x''(t) - 6x'(t) + 9x(t) = 9t + 3$$

2. 解微分方程

$$x''(t) + 16x(t) = 0$$

3. 一个抛物体在 20m 高的台子上以 20m/s 的速度被抛射出去, 速度与水平面的夹角

- 为 45° , 求它能达到的最大高度。
4. 求上题中的物体经过多长时间落地。
5. 一块石头从距离地面 50m 高处下落, 垂直地穿过一有阻力介质, 假设 $k = 1\text{s}^{-1}$, 用 5.1.4 节中介绍的牛顿方法估算石块落地的时间 t 。
6. 如图 10.6 所示, 在一个无摩擦的滑轮两端分别挂着一个物体, 其中悬挂物体的质量为 M , 另一物体的质量为 m , 它位于与地面夹角为 θ 的倾斜平面上。斜面的滑动摩擦系数为 μ_k , 假设 M 大于 m , 求悬挂物体下降的加速度 a 。提示: 两个物体都被加速, 并且作用于这个系统的所有外力之和应当为 $(M + m)a$ 。

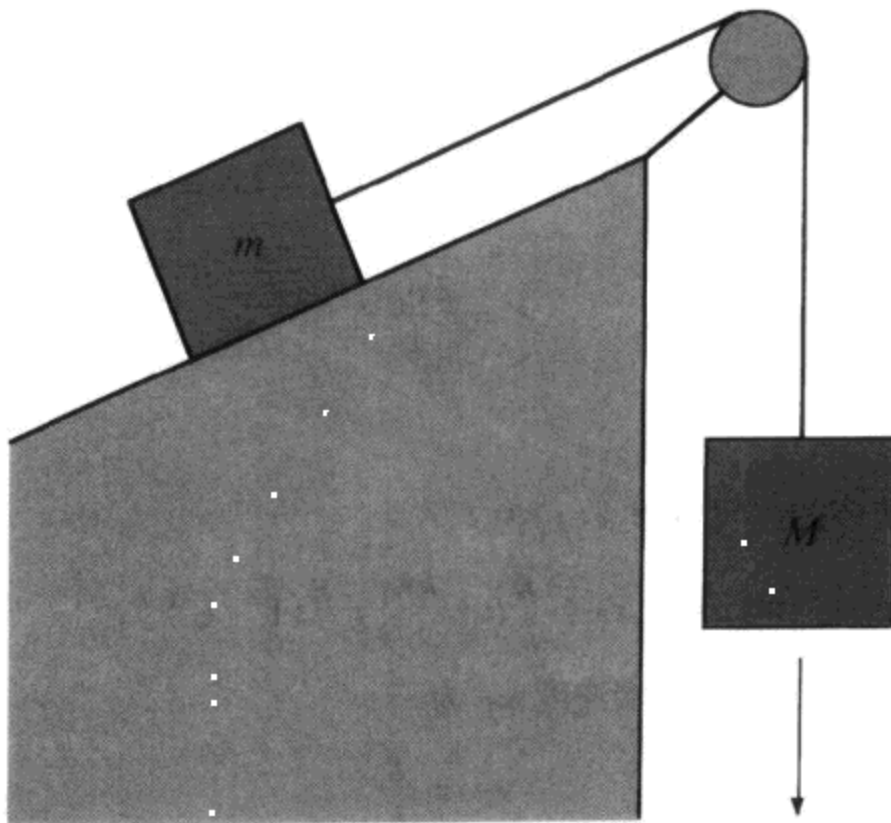
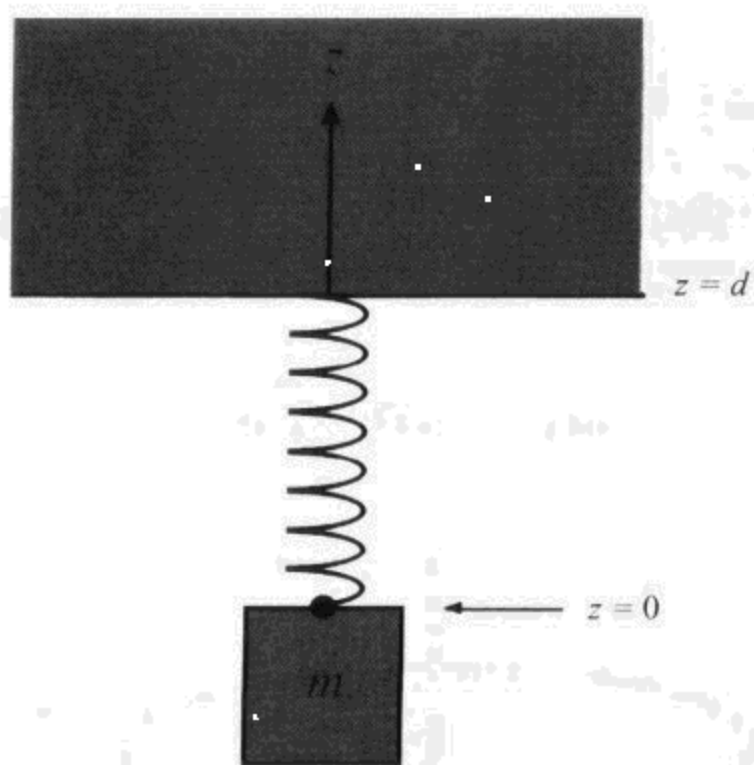


图 10.6 习题 6 中的滑轮

第 11 章 旋转物理运动



本章继续介绍计算机游戏编程中用到的物理方法。将进入旋转运动领域研究旋转物体的行为和作用于其上的力。旋转运动在游戏编程中有着广泛的应用,无论是模拟游戏中的角色与其他游戏对象之间交互作用,还是太空大战的仿真无不用到旋转运动的原理与方法。事实上,任何一个空中飞行的物体或不停在某个表面上的物体都可能作旋转运动,对这些物体的行为的准确模拟则非常有用。

11.1 旋转环境

本节讨论旋转环境中物体的物理学特性。旋转环境指的是任何绕着某一轴旋转的参照结构,包括儿童玩的旋转木马到人类赖以生存的地球。从介绍角速度开始,然后分析旋转环境中物体受到的各种力。

11.1.1 角速度

假设质量为 m 的质点正绕与单位向量 \mathbf{A} 平行的轴旋转,如图 11.1 所示,该质点被长度为 r 的线系在旋转轴上。令单位向量 \mathbf{X} 和 \mathbf{Y} 位于与向量 \mathbf{A} 垂直的平面上,并且向量 \mathbf{X} , \mathbf{Y} 和 \mathbf{A} 构成一个右手坐标轴系,也就是 $\mathbf{X} \times \mathbf{Y} = \mathbf{A}$ 。令 $\theta(t)$ 为在 t 时刻时,连接质点的线在 \mathbf{XY} 平面上的投影与 \mathbf{X} 轴之间沿逆时针方向的夹角。质点的角速度为角度 $\theta(t)$ 随时间的变化率,常用 ω 来表示

$$\omega(t) = \dot{\theta}(t) = \frac{d}{dt}\theta(t) \quad (11.1)$$

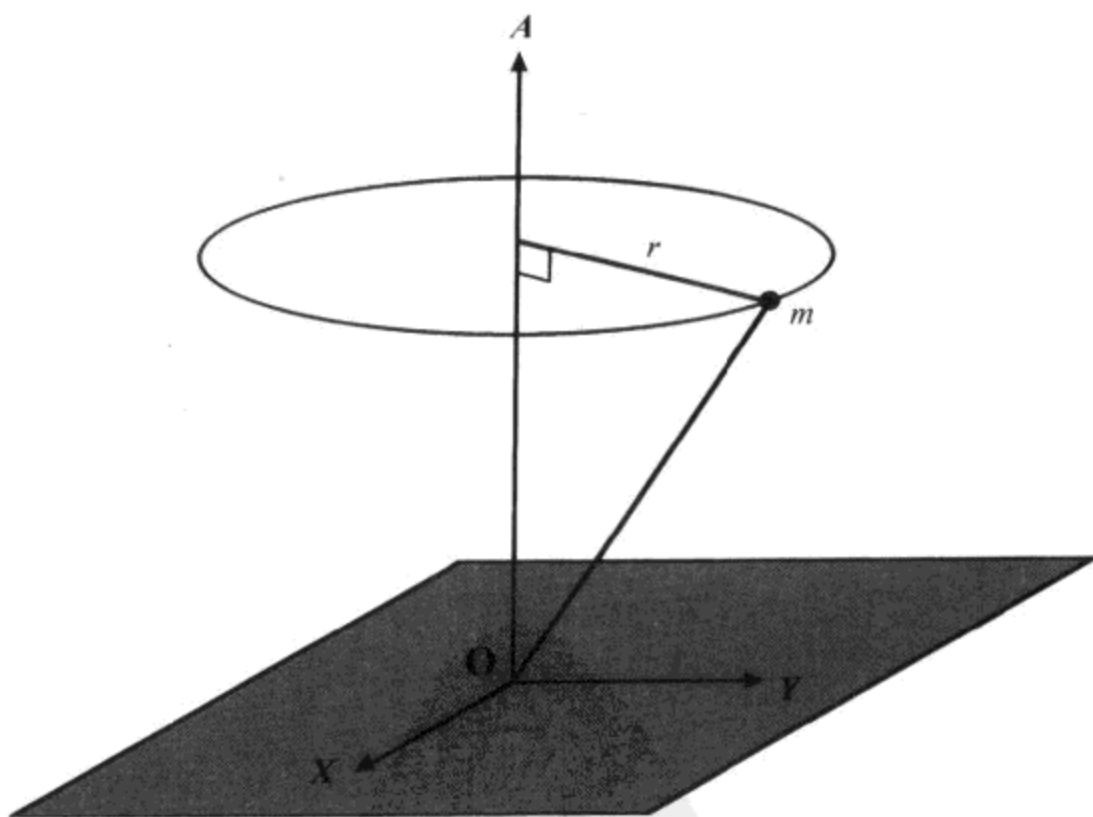


图 11.1 质点 m 绕与单位向量 \mathbf{A} 平行的轴旋转

角速度也常常用一个与旋转轴 \mathbf{A} 平行且大小为 $|\omega(t)|$ 的向量来表示,向量角速度 $\omega(t)$ 定义如下,即

$$\boldsymbol{\omega}(t) = \omega(t)\mathbf{A} = \dot{\theta}(t)\mathbf{A} \quad (11.2)$$

质点在空间中的运动速度(通常称为线速度)等于质点的角速度与其到旋转轴的距离(旋转半径)的积。图 11.1 中所示质点的线速度 $v(t)$ 就可以表示为

$$v(t) = |\omega(t)r| \quad (11.3)$$

然而,还是无法确定这个质点的运动方向。令向量函数 $r(t)$ 表示质点相对位于旋转轴上的固定原点的位置,如图 11.2 所示,质点的线速度向量 $v(t)$ 可表示为

$$\mathbf{v}(t) = \boldsymbol{\omega}(t) \times \mathbf{r}(t) \quad (11.4)$$

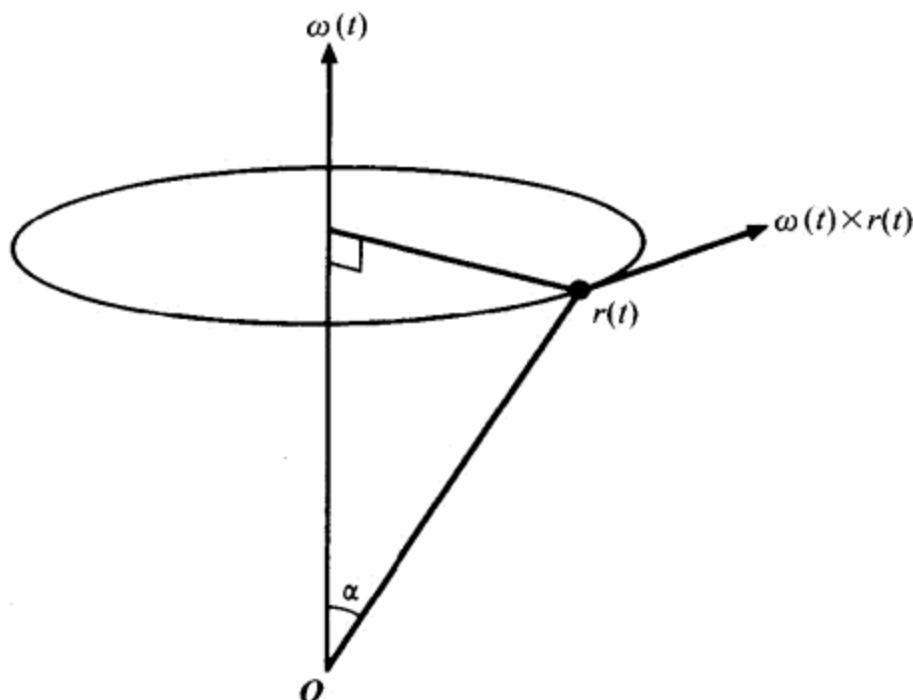


图 11.2 线速度向量 $v(t)$ 等于角速度向量 $\boldsymbol{\omega}(t)$ 与位置向量 $r(t)$ 的叉乘

这是因为质点到旋转轴的距离等于 $\|\mathbf{r}(t)\|$, 并且速度向量 $\mathbf{v}(t)$ 总是与指向旋转轴的方向垂直。

11.1.2 离心力

继续研究一个被绳系着的质点绕中心轴旋转的例子。已知质点的线性加速度 $\mathbf{a}(t)$ 为其线性速度对时间 t 的一阶导数, 计算方程(11.4)给出的线速度函数对时间的导数得

$$\mathbf{a}(t) = \dot{\mathbf{v}}(t) = \dot{\boldsymbol{\omega}}(t) \times \mathbf{r}(t) + \boldsymbol{\omega}(t) \times \dot{\mathbf{r}}(t) \quad (11.5)$$

因为 $\dot{\mathbf{r}}(t)$ 等于质点的线速度 $\mathbf{v}(t)$, 上式又可以写作

$$\mathbf{a}(t) = \dot{\boldsymbol{\omega}}(t) \times \mathbf{r}(t) + \boldsymbol{\omega}(t) \times [\boldsymbol{\omega}(t) \times \mathbf{r}(t)] \quad (11.6)$$

其中, 如果角速度为常量, 那么式 $\dot{\boldsymbol{\omega}}(t) \times \mathbf{r}(t)$ 的值为 0, 而 $\boldsymbol{\omega}(t) \times [\boldsymbol{\omega}(t) \times \mathbf{r}(t)]$ 项总是存在并且方向为从质点指向旋转中心。并且其方向将永远指向质点所绕的那根旋转轴的方向, 如图 11.3 所示。这部分加速度是由连接质点和旋转中心的绳的张力引起的。质点本身受到一个与绳子的张力大小相等而方向相反的力, 称为离心力。大家所熟知的使一个旋转物体远离或飞出旋转中心的力就是离心力, 其计算公式如下, 即

$$\mathbf{F}_{\text{离心力}} = -m(\boldsymbol{\omega}(t) \times [\boldsymbol{\omega}(t) \times \mathbf{r}(t)]) \quad (11.7)$$

当 $r(t)$ 和 $\boldsymbol{\omega}(t)$ 相互垂直时, 离心力可表示成标量的形式

$$F_{\text{离心力}} = m\omega^2 r = \frac{mv^2}{r} \quad (11.8)$$

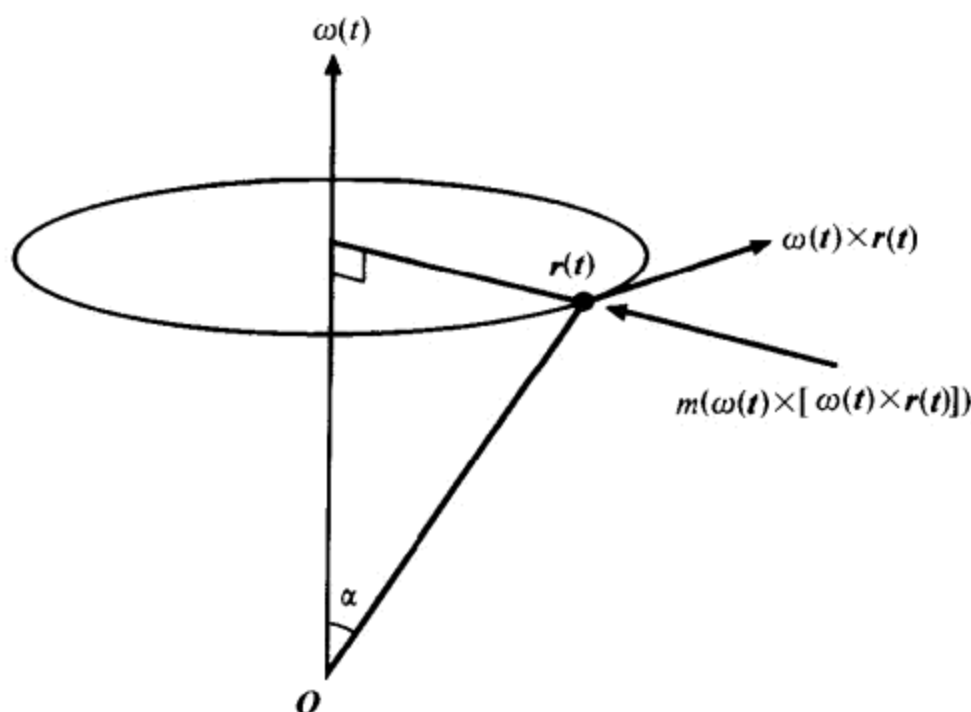


图 11.3 离心力

11.1.3 科里奥利力

现在讨论一个稍复杂的情况,即质点在一个转动着的物体表面上运动。假设一个质量为 m 的质点以角速度 $\omega(t)$ 绕某一轴旋转,并且该质点同时以速度 $v_r(t)$ 在一个旋转系统中运动,那么从旋转系统外的静止位置观察时,这个质点的速度为

$$v(t) = \omega(t) \times r(t) + v_r(t) \quad (11.9)$$

其中, $r(t)$ 是这个质点相对于旋转轴上某个原点的位置。因为速度 $v_r(t)$ 随着旋转系统一起旋转,那么对于一个静止观察者来说,质点相对固定坐标系的运动加速度为

$$a_f(t) = \omega(t) \times v_r(t) + a_r(t) \quad (11.10)$$

其中, $a_r(t) = \dot{v}_r(t)$ 是质点在旋转系统中的加速度。现在,可以得出这个质点的线性加速度 $a(t)$ 的完整公式为

$$\begin{aligned} a(t) &= \dot{v}_r(t) = \dot{\omega}(t) \times r(t) + \omega(t) \times \dot{r}(t) + a_f(t) \\ &= \dot{\omega}(t) \times r(t) + \omega(t) \times \dot{r}(t) + \omega(t) \times v_r(t) + a_r(t) \end{aligned} \quad (11.11)$$

由于 $\dot{r}(t)$ 等于这个质点的线性速度 $v(t)$, 因此上式又可以写为

$$a(t) = \dot{\omega}(t) \times r(t) + \omega(t) \times [\omega(t) \times r(t)] + 2\omega(t) \times v_r(t) + a_r(t) \quad (11.12)$$

那么,作用于质点上的力 $F(t)$ 可表示为

$$\begin{aligned} F(t) &= ma(t) = m\dot{\omega}(t) \times r(t) + m\omega(t) \times [\omega(t) \times r(t)] + \\ &\quad 2m\omega(t) \times v_r(t) + ma_r(t) \end{aligned} \quad (11.13)$$

在旋转系统中,作用于质点上的力为

$$\begin{aligned} F_r(t) &= ma_r(t) = F(t) - m\dot{\omega}(t) \times r(t) - m\omega(t) \times \\ &\quad [\omega(t) \times r(t)] - 2m\omega(t) \times v_r(t) \end{aligned} \quad (11.14)$$

正如所预料的,其中含有离心力,但除了离心力外还存在一个新的力,称为科里奥利力,它是

旋转系统中作用在质点上并与质点速度垂直的一种力,如图 11.4 所示,其计算公式为

$$F_{\text{科里奥利力}} = -2m\omega(t) \times v_r(t) \quad (11.15)$$

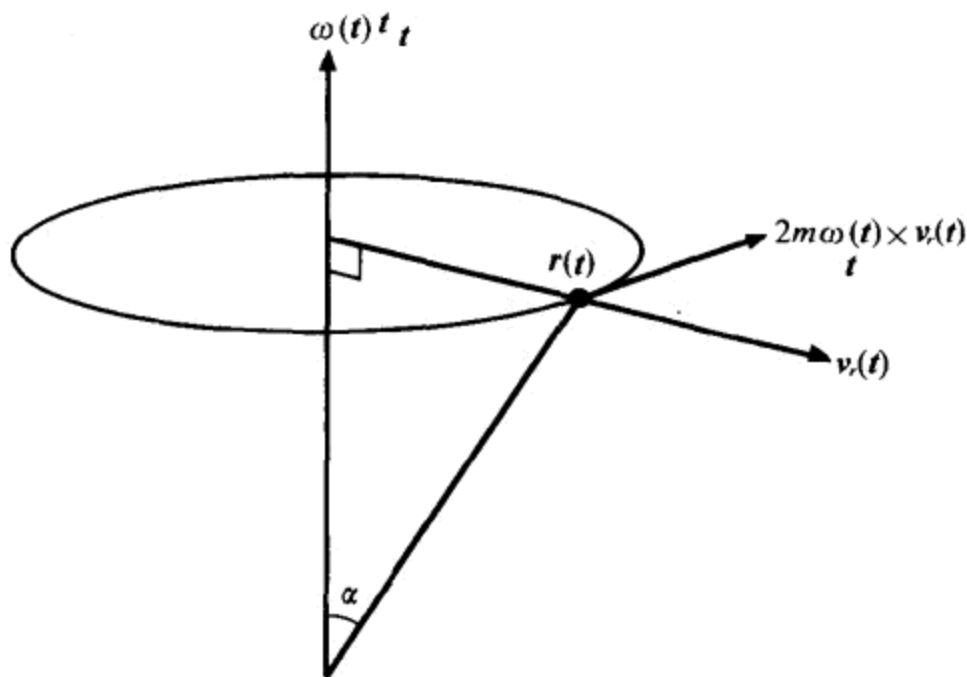


图 11.4 科里奥利力

科里奥利力只有当质点在一个旋转系统中运动时才会产生。某些气候现象中的大规模气旋运动就是由科里奥利力所引起的。例如,在北半球发生的飓风逆时针旋转而在南半球发生的飓风顺时针旋转,这是因为在赤道处方程(11.15)中的叉乘会改变符号。

11.2 刚体运动

刚体指的是由一系列紧密连接并有相同角速度的质点构成的物体。固体可以看作是由无穷多的质量无穷小的质点所构成的。因为构成刚体的质点之间没有相对运动,所以当刚体旋转时离心力和相对于其他质点位置是固定不变的。所以离心力和科里奥利力不起作用。刚体所能做的运动只有沿某个路径的线性运动或者绕某个中心轴的旋转运动。本节,将分析刚体的旋转运动的特性以及旋转中外力的作用效果。

11.2.1 质心

当一个刚体在没有任何外力作用自由旋转时,这个刚体其实就是绕着它的质量中心旋转。质量中心是刚体内部的点,如果以该点为原点,则作用于该点的任意方向的力都不会产生净扭矩,该点通常称为质心。

假设一个刚体是由多个质点组成,并且已知这些质点的位置和质量。那么这个刚体系统的总质量为

$$M = \sum_{\alpha} m_{\alpha} \quad (11.16)$$

其中, m_{α} 是构成刚体的第 α 个质点,显然将这些质点质量一一相加就可得到上式,用 r_{α} 表示第 α 个质点的位置,那么质心为

$$C = \frac{1}{M} \sum_a m_a r_a \quad (11.17)$$

对于一个固体来说,可以用以下的积分形式来计算它的总质量,即

$$M = \int_V dm(\mathbf{r}) \quad (11.18)$$

其中, $dm(\mathbf{r})$ 表示在位置 \mathbf{r} 处的质量微分, V 为这个固体的体积。令位置 \mathbf{r} 处的固体密度为 $\rho(\mathbf{r})$,则上式又可以写为

$$M = \int_V \rho(\mathbf{r}) dV \quad (11.19)$$

那么,固体的质心可用以下积分公式求出:

$$C = \frac{1}{M} \int_V \mathbf{r} \rho(\mathbf{r}) dV \quad (11.20)$$

【例 11.1】 计算一个半径为 R ,高度为 h ,密度为 ρ 的锥体的质心,这个锥体的底平面位于 xy 平面上,如图 11.5 所示。

解 利用柱面坐标系,该圆锥中离 xy 平面距离为 z 处的横截面的半径 $r(z)$ 为

$$r(z) = (h - z) \frac{R}{h} \quad (11.21)$$

利用方程(11.19)给出的积分公式来计算这个锥体的总质量,圆柱坐标系中体积的微分 dV 为

$$dV = r dr d\phi dz \quad (11.22)$$

因而,需要计算的积分公式为

$$M = \int_0^h \rho \pi [r(z)]^2 dz \quad (11.23)$$

其中,被积函数表示 xy 平面上方 z 处的圆盘的微分体积。用公式(11.21)替换上式中的 $r(z)$,得到

$$M = \rho \pi \frac{R^2}{h^2} \int_0^h (h - z)^2 dz = \frac{1}{3} \rho \pi R^2 h \quad (11.24)$$

由于锥体的圆周对称性,很显然质心的 x 和 y 分量为0,而质心的 z 分量可由方程(11.20)求得

$$C_z = \frac{1}{M} \int_0^h \int_0^{2\pi} \int_0^{r(z)} \rho z r dr d\phi dz \quad (11.25)$$

对 ϕ 求积分得

$$C_z = \frac{2\rho\pi}{M} \int_0^h \int_0^{r(z)} z r dr dz \quad (11.26)$$

再对 r 求积分,并用方程(11.21)给出的 $r(z)$ 的表达式代替 $r(z)$,得

$$C_z = \frac{\rho\pi R^2}{Mh^2} \int_0^h z (h - z)^2 dz = \frac{\rho\pi R^2}{Mh^2} \int_0^h (h^2 - 2hz^2 + z^3) dz \quad (11.27)$$

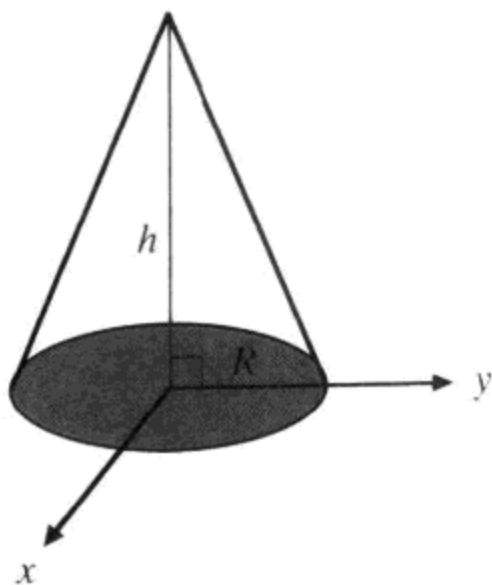


图 11.5 例 11.1 中的圆锥

最后,对 h 积分,可得

$$C_x = \frac{\rho\pi R^2 h^2}{12M} = \frac{h}{4} \quad (11.28)$$

这样就得到了锥体的质心位置 $C = \langle 0, 0, h/4 \rangle$ 。

11.2.2 角动量和扭矩

一个质量为 m 的质点以速度 \mathbf{v} 运动,其线性动量 \mathbf{p} 为 $m\mathbf{v}$ 。就像旋转运动中角速度与线性运动中的速度对应一样,旋转运动中与线性动量对应的量为角动量。

假设一个质量为 m 的质点,以角速度 $\boldsymbol{\omega}(t)$ 绕着某旋转中心旋转,这个质点的位置方程为 $\mathbf{r}(t)$,那么这个质点的角动量 $\mathbf{L}(t)$ 可定义为

$$\mathbf{L}(t) = \mathbf{r}(t) \times \mathbf{p}(t) \quad (11.29)$$

其中, $\mathbf{p}(t) = m\mathbf{V}(t)$ 是这个质点的线性动量。

对方程(11.29)两边的表达式求微分,可得

$$\dot{\mathbf{L}}(t) = \dot{\mathbf{r}}(t) \times \mathbf{p}(t) + \mathbf{r}(t) \times \dot{\mathbf{p}}(t) \quad (11.30)$$

因为 $\dot{\mathbf{r}}(t) = \mathbf{v}(t)$, 并且向量 $\dot{\mathbf{r}}(t)$ 和 $\mathbf{p}(t)$ 方向相同,所以叉乘 $\dot{\mathbf{r}}(t) \times \mathbf{p}(t)$ 的值为 0, 因此下式成立,即

$$\dot{\mathbf{L}}(t) = \mathbf{r}(t) \times \dot{\mathbf{p}}(t) = \mathbf{r}(t) \times m\dot{\mathbf{V}}(t) \quad (11.31)$$

向量 $m\dot{\mathbf{V}}(t)$ 等于作用于质点上的合力 $\mathbf{F}(t)$, 所以

$$\dot{\mathbf{L}}(t) = \mathbf{r}(t) \times \mathbf{F}(t) \quad (11.32)$$

方程(11.32)的右边式子叫做作用于质点的扭矩 $\boldsymbol{\tau}(t)$

$$\boldsymbol{\tau}(t) = \mathbf{r}(t) \times \mathbf{F}(t) \quad (11.33)$$

扭矩对应于线性作用力,可以产生角加速度。若作用于质点的净扭矩为 0, 那么角动量为保持不变,这是因为

$$\dot{\mathbf{L}}(t) = \boldsymbol{\tau}(t) \quad (11.34)$$

11.2.3 惯性张量

与角速度相关的角动量远比与线速度相关的线性动量复杂得多。事实上,角动量向量与角速度向量的方向并不一定相同,本节将讨论这两个量之间的关系。

由多个质点构成的刚体的角动量为

$$\mathbf{L}(t) = \sum_{\alpha} \mathbf{r}_{\alpha}(t) \times \mathbf{p}_{\alpha}(t) \quad (11.35)$$

其中, $\mathbf{r}_{\alpha}(t)$ 表示第 α 个质点的位置向量, $\mathbf{p}_{\alpha}(t)$ 表示第 α 个质点的动量,由于线性动量 $\mathbf{p}_{\alpha}(t)$ 可写为

$$\mathbf{p}_{\alpha}(t) = m_{\alpha}\mathbf{V}_{\alpha}(t) = m_{\alpha}\boldsymbol{\omega}(t) \times \mathbf{r}_{\alpha}(t) \quad (11.36)$$

所以角动量又可写为

$$\mathbf{L}(t) = \sum_{\alpha} m_{\alpha} \mathbf{r}_{\alpha}(t) \times [\boldsymbol{\omega}(t) \times \mathbf{r}_{\alpha}(t)] \quad (11.37)$$

由定理 1.9 的性质(6)给出的向量恒等式可知

$$\mathbf{P} \times (\mathbf{Q} \times \mathbf{P}) = \mathbf{P} \times \mathbf{Q} \times \mathbf{P} = p^2 \mathbf{Q} - (\mathbf{P} \cdot \mathbf{Q}) \mathbf{P} \quad (11.38)$$

所以,角动量也可以写为

$$\mathbf{L}(t) = \sum_a m_a (r_a^2(t) \boldsymbol{\omega}(t) - [\mathbf{r}_a(t) \cdot \boldsymbol{\omega}(t)] \mathbf{r}_a(t)) \quad (11.39)$$

暂时省略自变量 t , 角动量 \mathbf{L} 中的第 i 个分量为

$$L_i = \sum_a m_a \left[r_a^2 \omega_i - (\mathbf{r}_a)_i \sum_{j=1}^3 (\mathbf{r}_a)_j \omega_j \right] \quad (11.40)$$

其中, ω_i 又可以表示为

$$\omega_i = \sum_{j=1}^3 \omega_j \delta_{ij} \quad (11.41)$$

这里 δ_{ij} 称为克罗内克符号, 其定义可参考式(1.42)中的定义, 将克罗内克符号带入式(11.40)得

$$L_i = \sum_a m_a \sum_{j=1}^3 [r_a^2 \omega_j \delta_{ij} - (\mathbf{r}_a)_i (\mathbf{r}_a)_j \omega_j] = \sum_{j=1}^3 \omega_j \sum_a m_a [\delta_{ij} r_a^2 - (\mathbf{r}_a)_i (\mathbf{r}_a)_j] \quad (11.42)$$

那么, α 上的求和运算又可表示为 3×3 矩阵 \mathcal{J} 的第 (i, j) 个元素

$$\mathcal{J}_{ij} = \sum_a m_a [\delta_{ij} r_a^2 - (\mathbf{r}_a)_i (\mathbf{r}_a)_j] \quad (11.43)$$

利用上式, 角动量 \mathbf{L} 中的第 i 个分量可表示为

$$L_i = \sum_{j=1}^3 \omega_j \mathcal{J}_{ij} \quad (11.44)$$

这样, 角动量 $\mathbf{L}(t)$ 可写为

$$\mathbf{L}(t) = \mathcal{J} \boldsymbol{\omega}(t) \quad (11.45)$$

矩阵 \mathcal{J} 称为惯性张量, 它除了表示角速度与角动量之间的关系之外, 还表示作用于刚体上的扭矩 $\boldsymbol{\tau}(t)$ 与刚体的角加速度 $\mathbf{a}(t) = \dot{\boldsymbol{\omega}}(t)$ 之间的关系。对方程(11.45)求微分可得

$$\dot{\mathbf{L}}(t) = \boldsymbol{\tau}(t) = \mathcal{J} \mathbf{a}(t) \quad (11.46)$$

因此, 惯性张量可用以下 3×3 矩阵表示

$$\mathcal{J} = \begin{bmatrix} \sum_a m_a (r_a^2 - x_a^2) & - \sum_a m_a x_a y_a & - \sum_a m_a x_a z_a \\ - \sum_a m_a x_a y_a & \sum_a m_a (r_a^2 - y_a^2) & - \sum_a m_a x_a y_a \\ - \sum_a m_a x_a z_a & - \sum_a m_a y_a z_a & \sum_a m_a (r_a^2 - z_a^2) \end{bmatrix} \quad (11.47)$$

其中, $x_a = (\mathbf{r}_a)_1, y_a = (\mathbf{r}_a)_2$ 以及 $z_a = (\mathbf{r}_a)_3$, 显然, \mathcal{J} 是一个对称矩阵, 对角线元素 $\mathcal{J}_{11}, \mathcal{J}_{22}$ 和 \mathcal{J}_{33} 分别为相对于 x, y 和 z 轴的惯性矩, 非对角线的元素叫做惯性积。

对于一个连续的质量分布, 方程(11.43)可用如下积分式表示, 即

$$\mathcal{J}_{ij} = \int_V (\delta_{ij} r^2 - r_i r_j) dm(\mathbf{r}) \quad (11.48)$$

其中, $dm(\mathbf{r})$ 表示在位置 \mathbf{r} 处的微分质量, V 是刚体的体积。若用方程 $\rho(\mathbf{r})$ 表示位置 \mathbf{r} 处的密度, 则上式又可以表示为

$$\mathcal{J}_{ij} = \int_V (\delta_{ij} r^2 - r_i r_j) \rho(\mathbf{r}) dV \quad (11.49)$$

【例 11.2】 计算一个半径为 R , 球心位于原点并且有着均匀密度 ρ 的固体球相对于 z

轴的惯性矩。

解 关于 z 轴的惯性矩为惯性张量矩阵 \mathcal{J} 的元素, 首先计算积分

$$\mathcal{J}_{33} = \int_V (r^2 - z^2) \rho dV \quad (11.50)$$

其中, $r^2 - z^2$ 为球内一点到 z 轴距离的平方, 在球面坐标系中该值为 $r^2 \sin^2 \theta$, θ 为极坐标中的角度值。微分体积 dV 为

$$dV = r^2 \sin \theta dr d\theta d\phi \quad (11.51)$$

所以方程(11.50)又可以写为

$$\mathcal{J}_{33} = \int_0^{2\pi} \int_0^\pi \int_0^R (r^2 \sin^2 \theta) \rho r^2 \sin \theta dr d\theta d\phi = \rho \int_0^{2\pi} \int_0^\pi \int_0^R r^4 \sin^3 \theta dr d\theta d\phi \quad (11.52)$$

计算 r 和 ϕ 上的积分值, 得

$$\mathcal{J}_{33} = \frac{2}{5} \pi \rho R^5 \int_0^\pi \sin^3 \theta d\theta = \frac{2}{5} \pi \rho R^5 \int_0^\pi (1 - \cos^2 \theta) \sin \theta d\theta \quad (11.53)$$

用 $u = -\cos \theta$ 和 $du = \sin \theta d\theta$ 代替上式中相应项, 并求积分得

$$\mathcal{J}_{33} = \frac{2}{5} \pi \rho R^5 \int_{-1}^1 (1 - u^2) du = \frac{8}{15} \pi \rho R^5 \quad (11.54)$$

球体的体积公式为 $V = \frac{4}{3} \pi R^3$, 则所求惯性矩为

$$\mathcal{J}_{33} = \frac{2}{5} \rho V R^2 = \frac{2}{5} m R^2 \quad (11.55)$$

其中, $m = \rho V$ 为球体的质量。

考虑到球体的对称性, 那么关于 x 轴和 y 轴的惯性矩也等于 $\frac{2}{5} m R^2$ 。另外, 惯性积为 0, 所以惯性张量 \mathcal{J} 为

$$\mathcal{J} = \begin{bmatrix} 2mR^2/5 & 0 & 0 \\ 0 & 2mR^2/5 & 0 \\ 0 & 0 & 2mR^2/5 \end{bmatrix} \quad (11.56)$$

最后, 用惯性矩的标量形式 $\mathcal{J} = \frac{2}{5} m R^2$ 表示旋转球体的角动量, 即

$$\mathbf{L}(t) = \mathcal{J} \boldsymbol{\omega}(t) \quad (11.57)$$

【例 11.3】 计算一个均匀密度为 ρ , 高为 h 并且半径为 R 的固体圆柱的惯性张量, 该圆柱轴线与 z 轴重合, 并且中心位于坐标原点, 如图 11.6 所示。

解 首先, 计算圆柱相对于 z 轴的惯性矩, 这里利用圆柱坐标计算以下积分

$$\mathcal{J}_{33} = \int_V (s^2 - z^2) \rho dV \quad (11.58)$$

这里, 用 s^2 代表圆柱内一点到原点的距离, 以避免

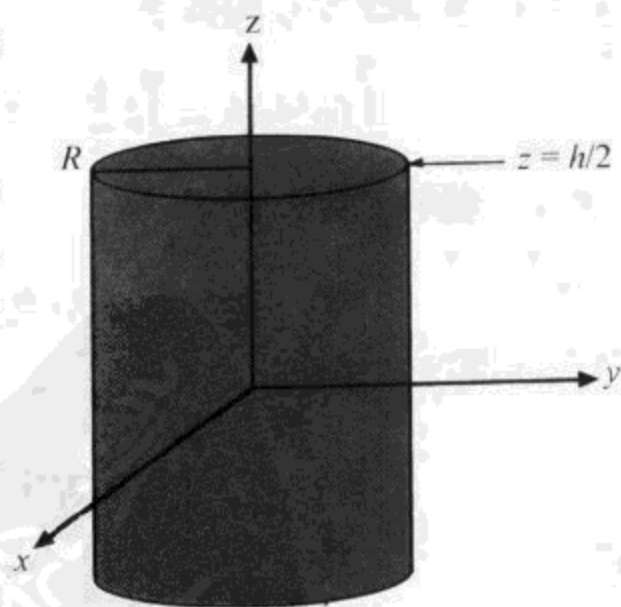


图 11.6 例 11.3 中的圆柱

与圆柱坐标系中的径向距离 r 混淆。 $s^2 - z^2$ 为该点到 z 坐标轴的距离,在圆柱坐标中即为 r^2 。在圆柱坐标系下微分体积 V 为

$$dV = r dr d\phi dz \quad (11.59)$$

于是,方程(11.58)为

$$J_{33} = \rho \int_{-h/2}^{h/2} \int_0^{2\pi} \int_0^R r^2 dr d\phi dz = \frac{1}{2} \pi \rho h R^4 \quad (11.60)$$

圆柱体的体积为 $V = \pi h R^2$, 所以相对 z 轴的惯性矩为

$$J_{33} = \frac{1}{2} \rho V R^2 = \frac{1}{2} m R^2 \quad (11.61)$$

其中, $m = \rho V$ 为柱体的质量。因为柱体关于 z 轴对称,所以 $J_{11} = J_{22}$,用以下积分计算关于 x 轴的惯性矩

$$J_{11} = \int_V (s^2 - x^2) \rho dV \quad (11.62)$$

将 $s^2 = r^2 + z^2$ 和 $x^2 = r^2 \cos^2 \phi$ 带入上式,得

$$\begin{aligned} J_{11} &= \rho \int_{-h/2}^{h/2} \int_0^{2\pi} \int_0^R (r^2 + z^2 - r^2 \cos^2 \phi) r dr d\phi dz \\ &= \rho \int_{-h/2}^{h/2} \int_0^{2\pi} \int_0^R r^3 \sin^2 \phi dr d\phi dz + \rho \int_{-h/2}^{h/2} \int_0^{2\pi} \int_0^R z^2 r dr d\phi dz \end{aligned} \quad (11.63)$$

对第一项中的 r 和 z 求积分,对第二项中的全部 3 个变量求积分得

$$J_{11} = \frac{1}{4} \rho h R^4 \int_0^{2\pi} \sin^2 \phi d\phi + \frac{1}{12} \pi \rho h^3 R^3 \quad (11.64)$$

利用附录 B.4 中给出的三角恒等式

$$\sin^2 \phi = \frac{1 - \cos 2\phi}{2} \quad (11.65)$$

可以计算出剩余的变量的积分

$$\int_0^{2\pi} \sin^2 \phi d\phi = \int_0^{2\pi} \left(\frac{1}{2} - \frac{1}{2} \cos 2\phi \right) d\phi = \frac{1}{2} \phi - \frac{1}{4} \sin 2\phi \Big|_0^{2\pi} = \pi \quad (11.66)$$

则,关于 x 轴和 y 轴的惯性矩为

$$J_{11} = J_{22} = \frac{1}{4} \pi \rho h R^4 + \frac{1}{12} \pi \rho h^3 R^3 = \frac{1}{4} m R^2 + \frac{1}{12} m h^2 \quad (11.67)$$

惯性积 J_{12} 等于以下积分

$$J_{12} = \int_V -xy \rho dV = -\rho \int_{-h/2}^{h/2} \int_0^{2\pi} \int_0^R r^3 \sin \phi \cos \phi dr d\phi dz \quad (11.68)$$

因为

$$\int_0^{2\pi} \sin \phi \cos \phi d\phi = 0 \quad (11.69)$$

所以 $\mathcal{I}_{12} = \mathcal{I}_{21} = 0$ 。可以证明其他惯性积也同样为 0, 所以圆柱体的惯性张量 \mathcal{I} 为

$$\mathcal{I} = \begin{bmatrix} \frac{1}{4}mR^2 + \frac{1}{12}mh^2 & 0 & 0 \\ 0 & \frac{1}{4}mR^2 + \frac{1}{12}mh^2 & 0 \\ 0 & 0 & \frac{1}{4}mR^2 \end{bmatrix} \quad (11.70)$$

其中, $m = \rho V$ 为柱体的质量。

非零的惯性积出现在当一个长方体绕通过它的一个顶点的轴旋转的情况下。在 11.2.4 一节中我们将专门分析惯性张量不为对角矩阵的情况。

【例 11.4】 计算一个均匀密度为 ρ , 一个顶点位于坐标原点并且三个边分别与坐标轴重合的长方体盒的惯性张量。该长方体盒的三个边长分别为 a, b 和 c , 如图 11.7 所示。

解 关于 x 轴的惯性矩 \mathcal{I}_{11} 的积分公式为

$$\begin{aligned} \mathcal{I}_{11} &= \int_V (r^2 - x^2) \rho dV \\ &= \rho \int_0^c \int_0^b \int_0^a (y^2 + z^2) dx dy dz \\ &= \rho ac \int_0^b y^2 dy + \rho ab \int_0^c z^2 dz \quad (11.71) \\ &= \frac{1}{3} \rho ab^3 c + \frac{1}{3} \rho abc^3 \\ &= \frac{1}{3} \rho abc (b^2 + c^2) \end{aligned}$$

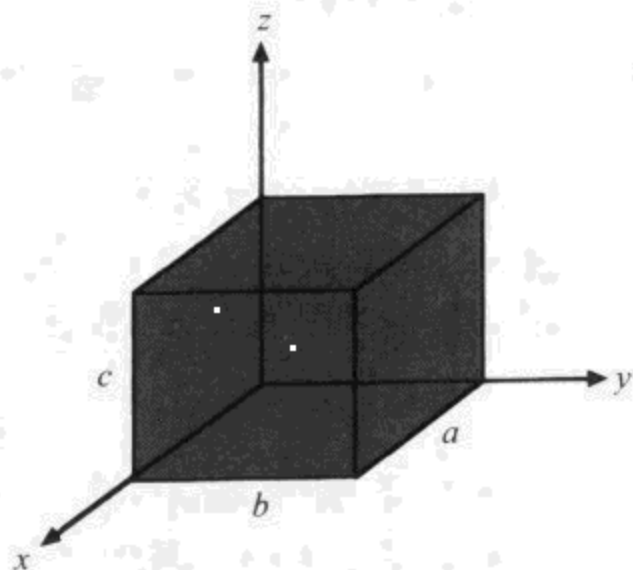


图 11.7 例 11.4 中的长方体盒

长方体的体积为 $V = abc$, 所以惯性矩为

$$\mathcal{I}_{11} = \frac{1}{3} m (b^2 + c^2) \quad (11.72)$$

其中, $m = \rho V$ 为盒子的质量。用同样的方法可以计算出立方体对 y 轴和 z 轴的惯性矩, 分别为

$$\mathcal{I}_{22} = \frac{1}{3} m (a^2 + c^2) \quad (11.73)$$

$$\mathcal{I}_{33} = \frac{1}{3} m (a^2 + b^2)$$

惯性积 \mathcal{I}_{12} 的积分公式为

$$\begin{aligned} \mathcal{I}_{12} &= - \int_V xy \rho dV = - \rho \int_0^c \int_0^b \int_0^a xy dx dy dz \\ &= - \rho \int_0^c \int_0^b xy dx dy \\ &= - \frac{1}{4} \rho a^2 b^2 c = - \frac{1}{4} m ab \quad (11.74) \end{aligned}$$

用同样的方法可得另外两个惯性积 \mathcal{I}_{13} 和 \mathcal{I}_{23} , 即

$$\begin{aligned} J_{13} &= -\frac{1}{4}mac \\ J_{23} &= -\frac{1}{4}mbc \end{aligned} \quad (11.75)$$

则长方体盒子的惯性张量为

$$J = \begin{bmatrix} \frac{1}{3}m(b^2 + c^2) & -\frac{1}{4}mab & -\frac{1}{4}mac \\ -\frac{1}{4}mab & \frac{1}{3}m(a^2 + c^2) & -\frac{1}{4}mbc \\ -\frac{1}{4}mac & -\frac{1}{4}mbc & \frac{1}{3}m(a^2 + b^2) \end{bmatrix} \quad (11.76)$$

这里, $m = \rho V$ 是长方体的质量。

11.2.4 主惯性轴

因为角动量 $L(t)$ 和角速度 $\omega(t)$ 之间的关系可表示为

$$L(t) = J\omega(t) \quad (11.77)$$

当 $\alpha(t)$ 是惯性张量 J 的特征向量时, 上式中的两个向量平行。因为惯性张量为对称矩阵, 所以它有三个实特征值, 并且相应的特征向量正交(见 2.5 节)。惯性张量的特征值被称为主惯性矩, 相应的特征向量为轴。若一个刚体绕着它的一个主惯性轴旋转, 则它的角动量为

$$L(t) = I\omega(t) \quad (11.78)$$

这里, I 为与该主惯性轴相对应的主惯性矩。

若惯性张量为一个对角矩阵, 例如球体或圆柱体的惯性张量, 则主惯性矩就为矩阵的对角元素且主惯性轴就是 x, y 和 z 坐标轴。若惯性张量不是对角矩阵, 则必须计算惯性张量矩阵的特征值和特征向量, 从而求出主惯性轴。

【例 11.5】 计算一个边长为 a , 有一个顶点位于坐标原点并且各边分别与坐标轴重合的立方体的主惯性轴。

解 立方体盒的惯性张量 J 可由方程(11.76)得出, 令方程中的三个边长都为 a , 就可求出立方体的惯性张量

$$J = \begin{bmatrix} \frac{2}{3}ma^2 & -\frac{1}{4}ma^2 & -\frac{1}{4}ma^2 \\ -\frac{1}{4}ma^2 & \frac{2}{3}ma^2 & -\frac{1}{4}ma^2 \\ -\frac{1}{4}ma^2 & -\frac{1}{4}ma^2 & \frac{2}{3}ma^2 \end{bmatrix} \quad (11.79)$$

其行列式为

$$\begin{bmatrix} \frac{2}{3}ma^2 - I & -\frac{1}{4}ma^2 & -\frac{1}{4}ma^2 \\ -\frac{1}{4}ma^2 & \frac{2}{3}ma^2 - I & -\frac{1}{4}ma^2 \\ -\frac{1}{4}ma^2 & -\frac{1}{4}ma^2 & \frac{2}{3}ma^2 - I \end{bmatrix} = 0 \quad (11.80)$$

相应的特征多项式的根即为惯性张量 \mathcal{J} 的特征值。根据行列式的性质可知将其中一行加到另一行上不会影响行列式的值, 这里将上述行列式的第 2 行减去第 1 行可以将其特征多项式简化为

$$\begin{bmatrix} \frac{2}{3}ma^2 - I & -\frac{1}{4}ma^2 & -\frac{1}{4}ma^2 \\ -\frac{11}{12}ma^2 + I & \frac{11}{12}ma^2 - I & 0 \\ -\frac{1}{4}ma^2 & -\frac{1}{4}ma^2 & \frac{2}{3}ma^2 - I \end{bmatrix} = 0 \quad (11.81)$$

将因子 $\frac{11}{12}ma^2 - I$ 从行列式第 2 行中提出, 并令 $b = -\frac{1}{4}ma^2$, 带入行列式, 则有

$$\left(\frac{11}{3}b - I\right) \begin{vmatrix} \frac{8}{3}b - I & -b & -b \\ -1 & 1 & 0 \\ -b & -b & \frac{8}{3}b - I \end{vmatrix} = 0 \quad (11.82)$$

计算最后的行列式, 可以得到

$$\begin{aligned} 0 &= \left(\frac{11}{3}b - I\right) \left[\left(\frac{8}{3}b - I\right)^2 - b\left(\frac{8}{3}b - I\right) - 2b^2 \right] \\ &= \left(\frac{11}{3}b - I\right) \left(I^2 - \frac{13}{3}bI + \frac{22}{9}b^2 \right) \\ &= \left(\frac{11}{3}b - I\right) \left(\frac{11}{3}b - I\right) \left(\frac{2}{3}b - I\right) \end{aligned} \quad (11.83)$$

则主惯性矩 I_1, I_2 和 I_3 为

$$\begin{aligned} I_1 &= \frac{11}{3}b = \frac{11}{12}ma^2 \\ I_2 &= \frac{11}{3}b = \frac{11}{12}ma^2 \\ I_3 &= \frac{2}{3}b = \frac{1}{6}ma^2 \end{aligned} \quad (11.84)$$

为了计算出特征值 I_3 对应的惯性主轴, 必须解以下齐次线性方程组, 即

$$\begin{bmatrix} \frac{2}{3}ma^2 - I_3 & -\frac{1}{4}ma^2 & -\frac{1}{4}ma^2 \\ -\frac{1}{4}ma^2 & \frac{2}{3}ma^2 - I_3 & -\frac{1}{4}ma^2 \\ -\frac{1}{4}ma^2 & -\frac{1}{4}ma^2 & \frac{2}{3}ma^2 - I_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{0} \quad (11.85)$$

再一次利用常量 $b = \frac{1}{4}ma^2$, 则 $I_3 = \frac{2}{3}b$, 带入上式得

$$\begin{bmatrix} 2b & -b & -b \\ -b & 2b & -b \\ -b & -b & 2b \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{0} \quad (11.86)$$

化简得线性方程组为

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{0} \quad (11.87)$$

解得 $x = y = z$, 也就是说, 向量 $\langle 1, 1, 1 \rangle$ 即立方体的对角线, 为与主惯性矩 I_3 对应的主惯性轴。与主惯性矩 I_1, I_2 对应的主惯性轴可以通过解以下方程组求得

$$\begin{bmatrix} \frac{8}{3} - I_1 & -b & -b \\ -b & \frac{8}{3} - I_1 & -b \\ -b & -b & \frac{8}{3} - I_1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{0} \quad (11.88)$$

化简得

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{0} \quad (11.89)$$

因此, 每一个主惯性轴的 y 和 z 分量只要不同时为 0, 它们可为任意值, x 分量的值则为 $x = -y - z$ 。由于任何形式为 $\langle -y - z, y, z \rangle$ 的向量都垂直于向量 $\langle 1, 1, 1 \rangle$, 与主惯性矩 I_1, I_2 对应的主惯性轴可以为与立方体对角线垂直的平面内的任意正交向量。

若一个刚体没有绕着它的主惯性轴旋转, 则角速度向量 $\boldsymbol{\omega}(t)$ 和角动量 $\mathbf{L}(t)$ 不平行, 在这种情况下, 向量 $\mathbf{L}(t)$ 绕着向量 $\boldsymbol{\omega}(t)$ 的方向旋转的速度为

$$\dot{\mathbf{L}}(t) = \boldsymbol{\omega}(t) \times \mathbf{L}(t) \neq \mathbf{0} \quad (11.90)$$

这样, 角加速度就会改变旋转轴。因为 $\dot{\mathbf{L}}(t) = \mathcal{J}\boldsymbol{\alpha}(t)$, 所以 $\boldsymbol{\alpha}(t)$ 为

$$\boldsymbol{\alpha}(t) = \mathcal{J}^{-1}\dot{\mathbf{L}}(t) = \mathcal{J}^{-1}[\boldsymbol{\omega}(t) \times \mathbf{L}(t)] \quad (11.91)$$

为了克服角速度的影响, 防止旋转轴的改变, 在角加速度的反方向上必须施加一个大小与向量 $\boldsymbol{\omega}(t) \times \mathbf{L}(t)$ 的大小一样的扭矩。因此, 一个旋转的刚体的旋转运动可用如下方程描述, 即

$$\sum_{i=1}^N \boldsymbol{\tau}_i(t) - \boldsymbol{\omega}(t) \times \mathbf{L}(t) = \mathcal{J}\boldsymbol{\alpha}(t) = \mathcal{J}\dot{\boldsymbol{\theta}}(t) \quad (11.92)$$

其中, $\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, \dots, \boldsymbol{\tau}_N$ 为作用于刚体的外部扭矩。该方程是与线性运动方程(10.9)对应的旋转运动方程。

11.3 振 动

一个物体若在一定时间周期内在一定空间区域内来回反复运动, 则这个物体的运动状态可称为振动。振动通常是由回复力引起的, 该力可能为常数也可能在运动物体远离平衡位置时增大。因为振动具有旋转运动的一些特征, 如角速度, 所以放到本章介绍。将在 11.3.2 节中介绍的钟摆运动, 就是一个物体绕着一点旋转并具备振动特性的例子。

11.3.1 弹性运动

如图 11.8 所示, 一个质量为 m 的物体挂在一个自然长度为 d 的弹簧的底部, 这时物体

就可能作振动。假设弹簧与 z 轴重合, 并且一端系于固定物体上, 固定物体的位置为 $z = d$, 而悬挂物体的质量落在另一端, 恰好与原点重合。不考虑重力对物体造成的影响, 当弹簧被拉伸或压缩时, 其长度也会相对弹簧的自然长度 d 进行延长或缩短, 此时弹簧产生的回复力就正比于悬挂物体相对自然位置的位移。如果悬挂物体沿 z 坐标轴的位置为 z , 则 z 轴方向上的回复力为

$$F = -kz \quad (11.93)$$

这个公式被称为虎克定律, 其中, 常量 k 是与弹簧的刚性对应的弹簧本身的特性, 称为弹性系数, 这个值越大就说明要使悬挂的物体偏离原点所须作的功越多。

对于悬挂于弹簧一端的物体, 其位置可以表示为时间 t 的函数。利用以下微分方程可以获得该位置函数

$$m\ddot{z}(t) = -kz(t) \quad (11.94)$$

上述方程的一般解为

$$z(t) = A\sin\omega t + B\cos\omega t \quad (11.95)$$

其中, $\omega = \sqrt{k/m}$ 为振动的角频率, 度量单位为“弧度每单位时间”。当单位时间为秒时, 每单位时间内的振动次数, 即频率 f 的单位为赫兹, 相应的角频率方程如下, 即

$$f = \frac{\omega}{2\pi} \quad (11.96)$$

振动周期 P 等于重复一个运动所需的时间, 可表示成频率的倒数, 即

$$P = \frac{2\pi}{\omega} = 2\pi\sqrt{\frac{m}{k}} \quad (11.97)$$

方程(11.95)中的常量 A 和 B 的具体数值由初始位置决定, 假设初始位置为 z_0 , 初始速度为 v_0 , 因为

$$\begin{aligned} z(0) &= B \\ \dot{z}(0) &= A\omega \end{aligned} \quad (11.98)$$

显然

$$\begin{aligned} A &= \frac{v_0}{\omega} \\ B &= z_0 \end{aligned} \quad (11.99)$$

用 10.2.1 节中介绍的方法, 方程(11.95)表示为

$$z(t) = C\sin(\omega t + \delta) \quad (11.100)$$

其中

$$\begin{aligned} C &= \sqrt{\frac{v_0^2}{\omega^2} + z_0^2} \\ \delta &= \sin^{-1} \frac{z_0}{C} \end{aligned} \quad (11.101)$$

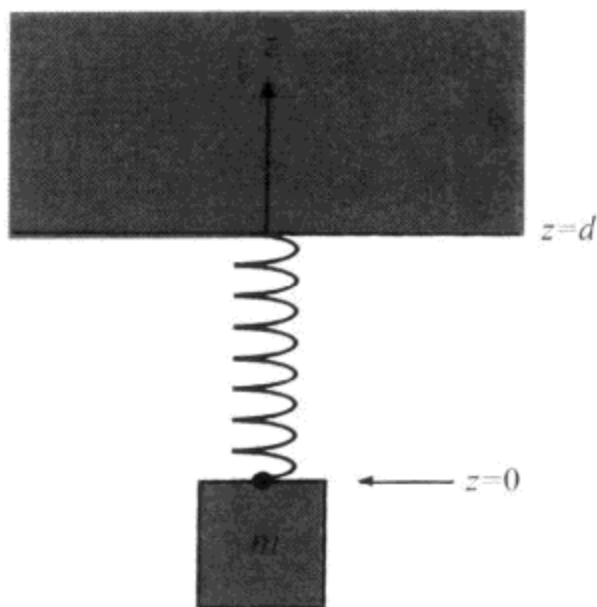


图 11.8 悬挂于弹簧一端的物体

常量 C 表示振动的幅度,为悬挂物体偏离平衡位置的最大距离。常量 δ 代表振动的相位,为悬挂物体的初始位置。

【例 11.6】 一个质量为 2kg 的物体系于弹性系数为 $k = 3\text{kg/s}^2$ 的弹簧的一端,求该物体振动的频率和振幅是多少? 假设物体已经位于平衡位置,并且在时间 $t = 0$ 时,从初始位置 $z_0 = -4\text{m}$ 处以初速度 $v_0 = -1\text{m/s}$ 被释放。

解 角频率 ω 为

$$\omega = \sqrt{\frac{k}{m}} = \frac{\sqrt{6}}{2} \text{rad/s} \quad (11.102)$$

除以 2π ,可以得到物体每秒振动的频率 f

$$f = \frac{\omega}{2\pi} = \frac{\sqrt{6}}{4\pi} \approx 0.915\text{Hz} \quad (11.103)$$

振动的幅度可用方程(11.101)求得

$$C = \sqrt{\frac{v_0^2}{\omega^2} + z_0^2} = \frac{5\sqrt{6}}{3} \approx 4.08\text{m} \quad (11.104)$$

现在考虑重力对物体振动的影响,假设质量为 m 的物体悬挂在竖直放置的弹簧一端,该物体受重力作用,将向下的重力常数 $-mg$ 加到方程(11.94)中,得

$$m\ddot{z}(t) = -kz(t) - mg \quad (11.105)$$

当弹簧的回复力与重力大小相对方向相反时,弹簧处于平衡状态,作用在它上面的合力为 0,则

$$-kz(t) = mg \quad (11.106)$$

解 $z(t)$ 可得悬挂重物的平衡位置

$$z(t) = -\frac{mg}{k} \quad (11.107)$$

若物体位于位置 $z = -mg/k$ 并且无初速度,则它将永远不会动。

方程(11.107)实际上是方程(11.105)的一个特解,将这个解代入方程(11.95)所示的齐次方程的一般解中,可得

$$z(t) = A\sin\omega t + B\cos\omega(t) - \frac{mg}{k} \quad (11.108)$$

同样代入相同的初始条件,即 $z(0) = z_0$ 和 $\dot{z}(0) = v_0$,可得到与方程(11.95)的常数 A 相同而 B 不同的解,即

$$A = \frac{v_0}{\omega} \quad (11.109)$$

$$B = z_0 + \frac{mg}{k}$$

将方程(11.108)写成以下形式:

$$z(t) = C\sin(\omega t + \delta) - \frac{mg}{k} \quad (10.110)$$

可见,振幅 C 和位置 δ 仍然保持不变,与式(11.101)相同。重力所起的作用仅是将弹簧拉长了距离 mg/k 。在重力作用下的振动仍然与没有重力影响的振动完全相同。

11.3.2 钟摆运动

假设在重力作用下,一个质量为 m 的物体悬挂在一根质量不计且长度为 L 的杆的一端,杆的另一端固定在一个位于坐标原点处的固定点上,如图 11.9 所示。继续假设这根杆可绕固定端自由摆动,悬挂物体也能在 xz 平面上移动。设 I 为物体相对于 y 轴(物体绕其旋转的轴)的惯性矩,那么,如果物体的质量都聚于一点,则有 $I = mL^2$ 。

令 $\mathbf{r}(t)$ 表示物体的位置,作用于物体的重力为 mg ,其产生的扭矩 $\boldsymbol{\tau}(t)$ 为

$$\boldsymbol{\tau}(t) = \mathbf{r}(t) \times mg \quad (11.111)$$

结果,角加速度 $\boldsymbol{\alpha}(t)$ 为

$$\boldsymbol{\alpha}(t) = \frac{\boldsymbol{\tau}(t)}{I} = \frac{\mathbf{r}(t) \times mg}{I} \quad (11.112)$$

因为 $\boldsymbol{\tau}(t)$ 和 $\boldsymbol{\alpha}(t)$ 总是垂直于钟摆运动的 xz 平面,可以将它们表示成标量的形式 $\tau(t)$ 和 $\alpha(t)$ 。方程(11.112)可写为

$$\alpha(t) = \ddot{\theta}(t) = -\frac{mgL}{I} \sin\theta(t) \quad (11.113)$$

这里 $\theta(t)$ 为钟摆与 z 轴负向之间的逆时针夹角。

由于方程 $\theta(t)$ 的正弦函数不一定存在,方程(11.113)不能用解析方法求解。然而,可以将 $\sin\theta(t)$ 用其幂级数的第一项来代替,从而对方程(11.113)进行求解

$$\ddot{\theta}(t) = -\frac{mgL}{I} \theta(t) \quad (11.114)$$

上式在角度 $\theta(t)$ 总是很小的情况下可近似表示钟摆的运动,其解为

$$\theta(t) = A \sin(\omega t + \delta) \quad (11.115)$$

其中,角频率 ω 为

$$\omega = \sqrt{\frac{mgL}{I}} \quad (11.116)$$

常量 A 和 δ 由初始位置决定,摆动的运动周期 P 为

$$P = \frac{2\pi}{\omega} = 2\pi \sqrt{\frac{I}{mgL}} \quad (11.117)$$

对于一个质量为单位质量的质点来说,有

$$\omega = \sqrt{\frac{g}{L}} \quad (11.118)$$

$$P = 2\pi \sqrt{\frac{L}{g}}$$

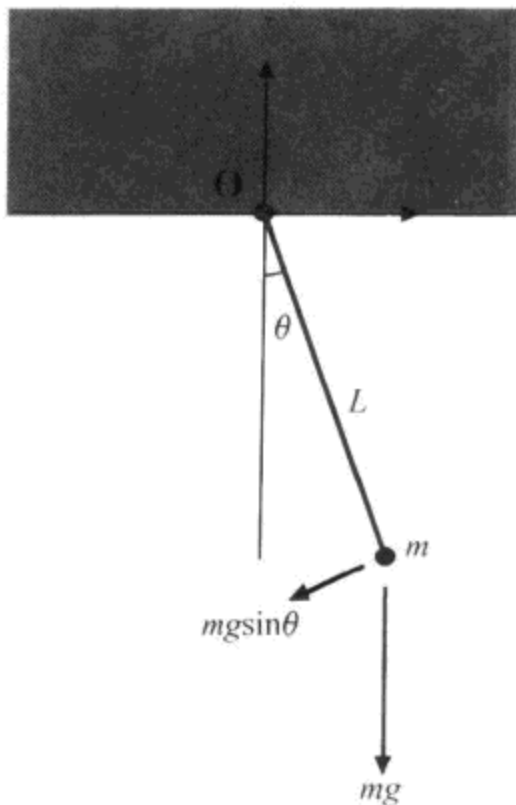


图 11.9 平面钟摆

11.4 本章小结

离心力

离心力是物体在旋转环境中受到的力,为

$$\mathbf{F}_{\text{离心力}} = -m(\boldsymbol{\omega}(t) \times [\boldsymbol{\omega}(t) \times \mathbf{r}(t)])$$

其中, $\boldsymbol{\omega}(t)$ 为角速度, $\mathbf{r}(t)$ 为物体相对于原点的位置。在 $\boldsymbol{\omega}(t)$ 和 $\mathbf{r}(t)$ 互相垂直的情况下, 离心力可表示成标量的形式

$$F_{\text{离心力}} = m\omega^2 r = \frac{mv^2}{r}$$

科里奥利力

科里奥利力也是一个物体在旋转系统中所受的力,为

$$\mathbf{F}_{\text{科里奥利力}} = -2m\boldsymbol{\omega}(t) \times \mathbf{V}_r(t)$$

其中, $\mathbf{V}_r(t)$ 是物体相对于旋转系统的速度。

质心

若一个固体在点 \mathbf{r} 处的密度为 $\rho(\mathbf{r})$, 该物体的质心 \mathbf{C} 为

$$\mathbf{C} = \frac{1}{M} \int_V \mathbf{r} \rho(\mathbf{r}) dV$$

其中, M 为该物体的总质量。

角动量

质点的角动量 $\mathbf{L}(t)$ 为

$$\mathbf{L}(t) = \mathbf{r}(t) \times \mathbf{p}(t)$$

其中, $\mathbf{r}(t)$ 为物体相对原点的位置, 并且物体的旋转轴通过该原点, $\mathbf{p}(t) = m\mathbf{V}(t)$ 为质点的线性动量。

扭矩

作用于质点的扭矩 $\boldsymbol{\tau}(t)$ 为

$$\boldsymbol{\tau}(t) = \mathbf{r}(t) \times \mathbf{F}(t)$$

其中, $\mathbf{F}(t)$ 为作用于位置 $\mathbf{r}(t)$ 处的力。作用于质点的扭矩之和为质点的相对时间的变化率

$$\dot{\mathbf{L}}(t) = \boldsymbol{\tau}(t)$$

惯性张量

一个刚体的惯性张量矩阵 \mathcal{I} 的元素 (i, j) 为

$$I_{ij} = \int_V (\delta_{ij} r^2 - r_i r_j) \rho(\mathbf{r}) dV$$

其中, $\rho(r)$ 为物体在 r 处的密度。惯性张量将角速度与角动量以及角加速度与扭矩联系起来, 即

$$\begin{aligned} \mathbf{L}(t) &= \mathcal{J}\boldsymbol{\omega}(t) \\ \boldsymbol{\tau}(t) &= \mathcal{J}\boldsymbol{\alpha}(t) \end{aligned}$$

弹性运动

质量为 m 的物体悬挂在弹性系数为 k 的弹簧一端, 其位置 $z(t)$ 为

$$z(t) = C\sin(\omega t + \delta)$$

其中

$$\begin{aligned} C &= \sqrt{\frac{mV_0^2}{R} + z_0^2} \\ \delta &= \sin^{-1} \frac{z_0}{C} \end{aligned}$$

这里, z_0 为初始位置, v_0 为初始速度。

钟摆运动

钟摆系统由一个质量为 m 的物体固定在长为 L 的杆上而构成, 它遵循以下运动方程:

$$\alpha(t) = \ddot{\theta}(t) = -\frac{mgL}{I}\sin\theta(t)$$

其中, $\theta(t)$ 是物体与竖直方向之间的夹角, I 是惯性矩。小角度钟摆运动的角频率为

$$\omega = \sqrt{\frac{mgL}{I}}$$

习 题

1. 一只蚂蚁以速度 v 在一个逆时针转动的圆盘上沿半径方向向外爬行, 圆盘的角速度为 ω , 在距离圆盘中心 r 处, 作用于这只蚂蚁的力是多少?
2. 假设一个质量为 m 的块状物位于一个转动的圆盘上, 它与圆盘的旋转中心之间的距离为 r , 若圆盘表面的静摩擦系数为 μ_S , 计算使块状物向外滑动, 圆盘最小的旋转角速度应为多大?
3. 计算一个半径为 R , 高度为 h 并且底面中心位于 xy 平面原点的圆柱的质心 C 。圆柱的密度为 $\rho(r) = 1 + r_z/h$ 。
4. 如图 11.10 所示, 计算内半径为 R_1 , 外半径为 R_2 , 高为 h 的圆柱环相对 z 轴的惯性矩。圆柱环的质量为 m 并且均匀密度为 ρ 。
5. 对于一个 x , y 和 z 方向的边长分别为 a , b 和 c 的长方体, 计算该长方体关于质心的惯性矩。长方体的质量为 m , 均匀密度为 ρ 。
6. 假设一个质量为 m 的物体被一根质量不计的绳子系于一个圆柱体的卷轴上, 并且绳子绕了很多圈, 如图 11.11 所示。若圆柱体的质量为 M , 半径为 R , 求在重力的作用下, 这个物体下降的加速度 a 。假设绳子不会从卷轴上脱落。(提示: 因为重力

作用于物体,所以绳子上将会产生一个张力 T ,该力作用于圆柱卷轴上,因此有方程 $ma = mg - T$ 。张力 T 会在柱体上产生一个扭矩,从而产生角加速度 α , $a = R\alpha$ 。)

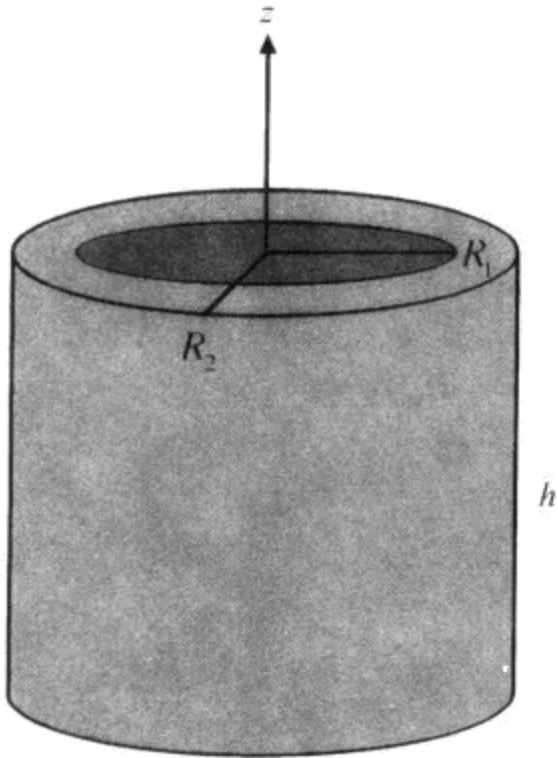


图 11.10 习题 4 中的圆柱环

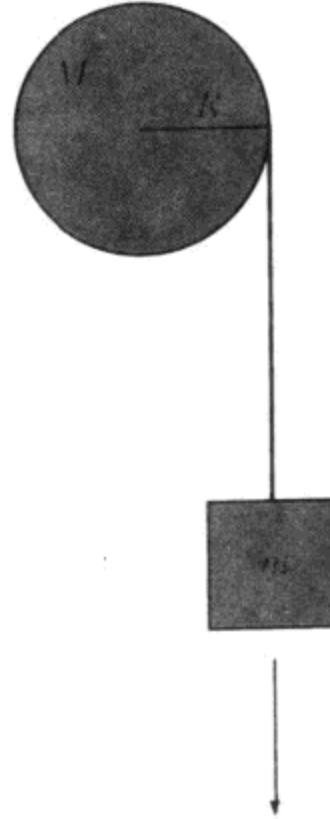


图 11.11 习题 6 中的受力系统

7. 一个质量为 m , 半径为 R 的球置于一个斜面上, 斜面与水平地面的夹角为 θ , 如图 11.12 所示。倾斜表面的静摩擦系数为 μ_s , 如果球体在重力作用下从斜面上滚下 (没有滑动), 计算球体的加速度 a 。(提示: 有两个力作用于物体上, 即重力和摩擦力, 它们之和为 ma , 摩擦力同时在球体上产生了一个扭矩, 从而产生了一个角加速度 α , $a = R\alpha$ 。)
8. 假设一个质量为 m 密度均匀的长方体盒位于地面上, 如图 11.13 所示。地面的静摩擦系数为 μ_s , 盒子的底部为长和宽均为 d 的正方形, 高为 h 。求作用在盒子一侧水平中心的力 $F < \mu_s mg$ 将盒子推翻的最小高度 z 。(提示: 使作用在质心的重力产生的扭矩等于力 F 相对盒子的一个底边的扭矩, 该底边为盒子中与力 F 的作用点相对的一侧的底边。)

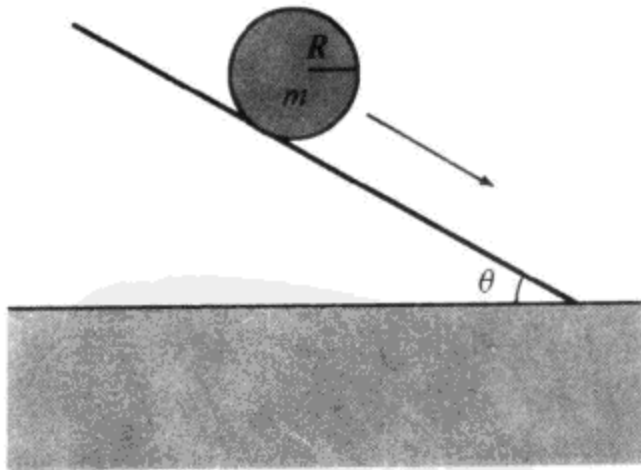


图 11.12 习题 7 中的球体

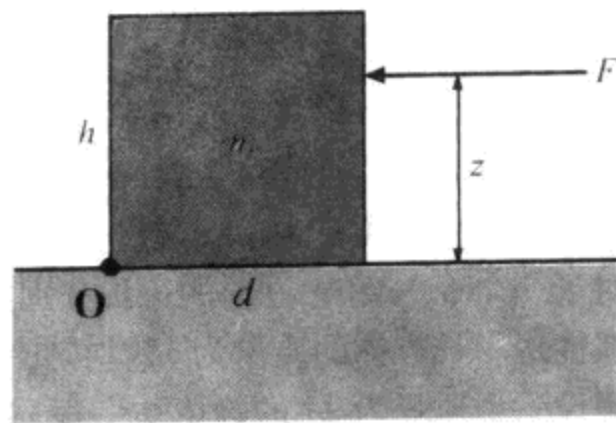
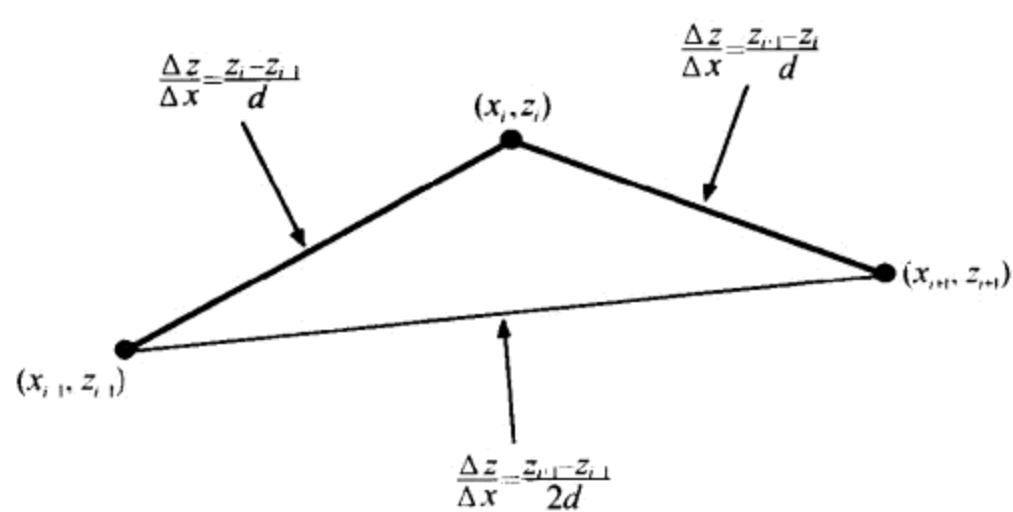


图 11.13 习题 8 中的长方体盒

第 12 章 流体仿真



许多游戏的场景中含有大量被流体覆盖的表面,这些流体覆盖的表面可能是一个水池、一桶强酸或者山沟里的熔岩。我们也希望游戏中的这些流体表面和现实物理世界中的一样,为了实现这些效果,需要模拟出像波通过流体表面而杂乱蔓延的效果。本章介绍著名的波方程并且用它进行流体表面的实时仿真。

12.1 波方程

波方程是一个偏微分方程,它描述在恒定张力作用下的一维直线或者二维表面上点的运动。如图 12.1 所示,一条有弹性的绳子紧紧绑在固定的两个端点之间,这两个端点都位于 x 坐标轴上,可以推导出二维波方程。假设绳子的线性密度 ρ 均匀(每单位长度的质量相等),同时受到一个切向的恒定张力 T 。

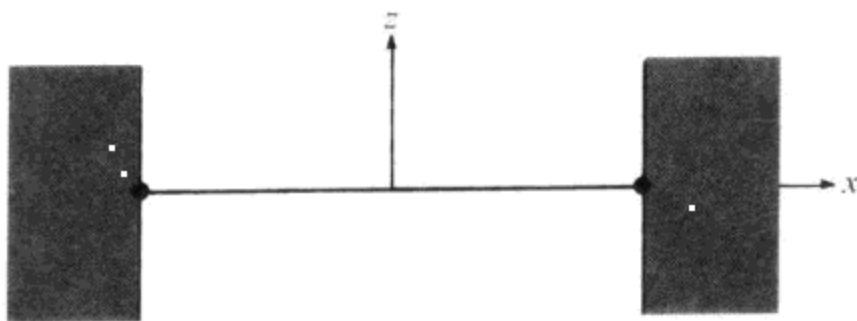


图 12.1 固定于两点之间线性密度为 ρ 的绳子

令函数 $z(x, t)$ 表示绳子在时间 t 时刻,水平位置为 x 处的竖直位移,当绳子在 z 方向发生位移时,绳子的拉伸将在绳子上每一点处产生一个力,该力将使绳子上的该点产生一个加速度。此时,牛顿第二定律表明某个时刻 t 作用在位于 $x = s$ 与 $x = s + \Delta x$ 之间的一小段绳子上的净受力 $F(x, t)$ 等于该段绳子的质量和相应加速度 $a(x, t)$ 之积,由于绳子线性密度为 ρ ,那么这段绳子的质量为 $\rho\Delta x$,则有

$$a(x, t) = \frac{F(x, t)}{\rho\Delta x} \quad (12.1)$$

如图 12.2 所示,可以把位于 $x = s$ 与 $x = s + \Delta x$ 之间的每一端点的受力分为水平力 $H(s, t)$ 和垂直力 $V(s, t)$ 。令 θ 表示在 $x = s$ 端点处 x 轴与绳子切线之间的角,由于张力 T 的方向沿绳子的切线方向,则张力的水平分量 $H(s, t)$ 与垂直分量 $V(s, t)$ 的方程如下,即

$$\begin{aligned} H(s, t) &= T \cos\theta \\ V(s, t) &= T \sin\theta \end{aligned} \quad (12.2)$$

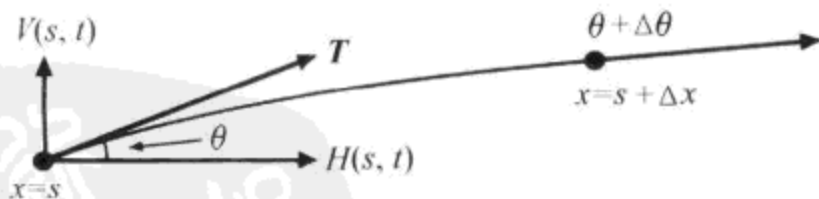


图 12.2 力能被分为水平与垂直两个分量

令 $\theta + \Delta\theta$ 表示端点 $x = s + \Delta x$ 处绳子的切线与 x 轴的夹角,则该端点处张力的水平分

力 $H(s + \Delta x, t)$ 和垂直分力 $V(s + \Delta x, t)$ 的方程为

$$\begin{aligned} H(s + \Delta x, t) &= T \cos(\theta + \Delta\theta) \\ V(s + \Delta x, t) &= T \sin(\theta + \Delta\theta) \end{aligned} \quad (12.3)$$

对于幅度较小的运动而言,假定张力的水平分量为0,则绳段只沿着竖直方向作加速运动,从而,对于位于 $x = s$ 与 $x = s + \Delta x$ 之间绳段,可得

$$H(s + \Delta x, t) - H(s, t) = 0 \quad (12.4)$$

于是,函数 H 与 x 无关,可用 $H(t)$ 来代替 $H(x, t)$ 。

作用于 $x = s$ 与 $x = s + \Delta x$ 之间的绳段上的张力的垂直分力产生的加速度可用方程(12.1)的 z 分量表示。由于垂直加速度为位移函数 $z(x, t)$ 的二阶导数,则

$$a_z(s, t) = \frac{\partial^2}{\partial t^2} z(s, t) = \frac{V(s + \Delta x, t) - V(s, t)}{\rho \Delta x} \quad (12.5)$$

两边同乘以 ρ ,再令 $\Delta x \rightarrow 0$,得

$$\rho \frac{\partial^2}{\partial t^2} z(s, t) = \lim_{\Delta x \rightarrow 0} \frac{V(s + \Delta x, t) - V(s, t)}{\Delta x} \quad (12.6)$$

上式右边等于速度 V 在 $x = s$ 时的偏导数,这样,等式(12.6)也可写为

$$\rho \frac{\partial^2}{\partial t^2} z(s, t) = \frac{\partial}{\partial x} V(s, t) \quad (12.7)$$

利用方程(12.2)给出 $V(s, t)$ 与 $H(t)$ 的值, $V(s, t)$ 可以表示为 $H(t)$ 的函数,即

$$V(s, t) = H(t) \tan\theta \quad (12.8)$$

由于 θ 为绳的切线与 x 轴之间的夹角, θ 的正切函数为函数 $z(x, t)$ 在 $x = s$ 处的斜率,则

$$V(s, t) = H(t) \frac{\partial}{\partial x} z(s, t) \quad (12.9)$$

同时方程(12.7)可写为

$$\rho \frac{\partial^2}{\partial t^2} z(s, t) = \frac{\partial}{\partial x} \left[H(t) \frac{\partial}{\partial x} z(s, t) \right] \quad (12.10)$$

由于 $H(t)$ 与 x 无关,上式又可写为

$$\rho \frac{\partial^2}{\partial t^2} z(s, t) = H(t) \frac{\partial^2}{\partial x^2} z(s, t) \quad (12.11)$$

在振幅很小的运动中, $\cos\theta$ 几乎等于1,可以用张力 T 近似表示 $H(t)$ 。令 $c^2 = T/\rho$,则一维波方程为

$$\frac{\partial^2 z}{\partial t^2} = c^2 \frac{\partial^2 z}{\partial x^2} \quad (12.12)$$

通过给方程(12.12)加一个二阶偏导数项可得二维波方程,即

$$\frac{\partial^2 z}{\partial t^2} = c^2 \left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right) \quad (12.13)$$

常数 c 表示每单位时间位移的距离,可代表速度。事实上,在这里不能证明 c 是实际上波传过一个平面或一条直线的速度。波速随传播介质受的张力增大而增大,而随着介质密度的增加而减小。

方程(12.13)中除了表面张力之外再无其他的力,这样,在整个表面上波的平均振幅不

会向现实世界中那样逐渐消失。可以为方程(12.13)加上一个粘性阻力,该力的方向与表面上点的速度方向相反,于是得

$$\frac{\partial^2 z}{\partial t^2} = c^2 \left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right) - \mu \frac{\partial z}{\partial t} \quad (12.14)$$

其中,非负常数 μ 代表流体的粘性, μ 的值大体上决定了波在一种表面上传播的距离,也就是波消失的时间。较小的 μ 值就可以让波存在时间较长,如水波。较大 μ 值会导致波的快速消失,如浓稠的油波。

12.2 近似导数

式(12.14)中所示的含粘性阻尼力的二维波方程,可以用分离变量的解析方法求解,但它的求解过程十分复杂,在实时仿真时需要相当大的计算量。为了克服解析方法的这个缺点,用数值方法模拟波在流体表面的传播过程。

如图 12.3 所示,假设流体表面为一个三角形网格,它的顶点排列成 $n \times m$ 的规则栅格。令 d 表示栅格中 x 和 y 方向上相邻顶点之间的距离, t 表示流体的连续两个状态之间的时间间隔。用 $z(i, j, k)$ 表示流体表面任一点的位移,其中 i 和 j 是分别满足条件 $0 \leq i \leq n$ 和 $0 \leq j \leq m$ 的整数,表示空间位置坐标, k 为非负整数,代表单元的时间坐标,则 $z(i, j, k)$ 即为位于 $\langle id, jd \rangle$ 处的顶点在时间 kt 时的位移函数。

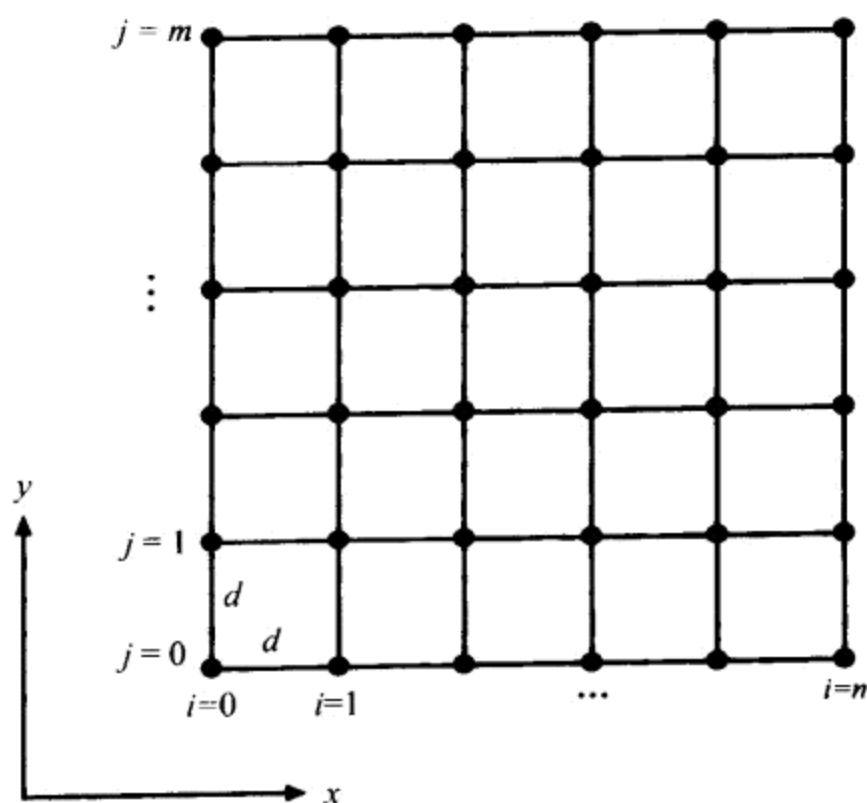
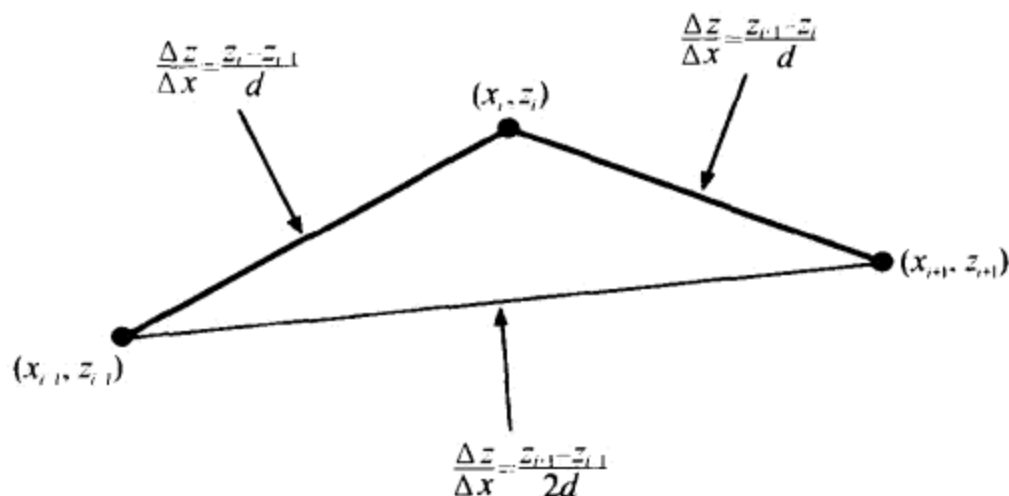


图 12.3 用三角网格表示的流体表面的顶点排列成一个 $n \times m$ 栅格

利用边界条件,即流体表面边界上顶点位移为 0,则表面内部点的位移可以用方程(12.14)计算,该方程中的导数可用相邻顶点之间的位移差近似表示。如图 12.4 所示,通过计算坐标为 (i, j) 的顶点与它在 x 方向上两个相邻顶点之间的 Δz 与 Δx 比值的平均值,可以近似得到在该点处表面的 x 方向的切向量。已知 $\Delta x = d$,则导数 $\partial z / \partial x$ 的定义如下,

$$\begin{aligned}\frac{\partial}{\partial x}z(i,j,k) &= \frac{\frac{z(i,j,k) - z(i-1,j,k)}{d} + \frac{z(i+1,j,k) - z(i,j,k)}{d}}{2} \\ &= \frac{z(i+1,j,k) - z(i-1,j,k)}{2d}\end{aligned}\quad (12.15)$$

图 12.4 近似得到在该点处表面的 x 方向的切向量

同理,通过计算坐标为 (i,j) 的顶点与它在 y 方向上两个相邻顶点之间的 Δz 与 Δy 比值的平均值,可以近似得到在该点处导数 $\partial z/\partial y$ 的值。已知 $\Delta y = d$,则导数 $\partial z/\partial y$ 的定义如下:

$$\frac{\partial}{\partial y}z(i,j,k) = \frac{z(i,j+1,k) - z(i,j-1,k)}{2d}\quad (12.16)$$

时间导数 $\partial z/\partial t$ 可定义为计算某个顶点在当前时刻与前一时刻的位移差和当前时刻与下一时刻的位移差的平均值。又时间间隔为 t ,于是 Δz 与 Δt 之比的平均值为

$$\frac{\partial}{\partial t}z(i,j,k) = \frac{z(i,j,k+1) - z(i,j,k-1)}{2t}\quad (12.17)$$

二阶导数可以用计算一阶导数的近似值的同样方法获得,即可以通过计算时间坐标或空间坐标中的一阶导数差的平均值求出二阶导数。为了说明求解过程,计算坐标为 (i,j) 的顶点处相对 x 的二阶导数。该顶点处一阶导数的平均差 $\Delta(\partial z/\partial x)$ 为

$$\Delta\left[\frac{\partial}{\partial x}z(i,j,k)\right] = \frac{\frac{\partial}{\partial x}z(i+1,j,k) - \frac{\partial}{\partial x}z(i-1,j,k)}{2}\quad (12.18)$$

将由方程(12.15)得出的相对 x 的导数值带入上式得

$$\begin{aligned}\Delta\left[\frac{\partial}{\partial x}z(i,j,k)\right] &= \frac{\frac{z(i+2,j,k) - z(i,j,k)}{2d} - \frac{z(i,j,k) - z(i-2,j,k)}{2d}}{2} \\ &= \frac{z(i+2,j,k) - 2z(i,j,k) + z(i-2,j,k)}{4d}\end{aligned}\quad (12.19)$$

除以 d 可得 $\Delta(\partial z/\partial x)$ 与 Δx 之比,即为二阶导数的近似值

$$\frac{\partial^2}{\partial x^2} z(i, j, k) = \frac{z(i+2, j, k) - 2z(i, j, k) + z(i-2, j, k)}{4d^2} \quad (12.20)$$

可以看出,在某个顶点处二阶导数的计算过程中要用到与该顶点距离两个顶点间距处的顶点位移。幸运的是,计算过程中没有用到与该顶点相邻的两个顶点,只要将坐标系以 (i, j) 顶点为中心缩小 $1/2$,同时 Δx 也缩小 $1/2$,就可以获得对于 x 的二阶导数的等价近似方程,即

$$\frac{\partial^2}{\partial x^2} z(i, j, k) = \frac{z(i+1, j, k) - 2z(i, j, k) + z(i-1, j, k)}{d^2} \quad (12.21)$$

下面的公式分别给出了相对空间坐标 y 和时间坐标 t 的二阶导数的近似值

$$\frac{\partial^2}{\partial y^2} z(i, j, k) = \frac{z(i, j+1, k) - 2z(i, j, k) + z(i, j-1, k)}{d^2} \quad (12.22)$$

$$\frac{\partial^2}{\partial t^2} z(i, j, k) = \frac{z(i, j, k+1) - 2z(i, j, k) + z(i, j, k-1)}{t^2} \quad (12.23)$$

12.3 计算表面位移

利用方程(12.17)中给出相对变量 t 的一阶导数以及由方程(12.21)、方程(12.22)和方程(12.23)给出的二阶导数,方程(12.14)中给出含粘性阻力的二维波方程在顶点 (i, j) 处的表达式为

$$\begin{aligned} & \frac{z(i, j, k+1) - 2z(i, j, k) + z(i, j, k-1)}{t^2} = \\ & c^2 \frac{z(i+1, j, k) - 2z(i, j, k) + z(i-1, j, k)}{d^2} + \\ & c^2 \frac{z(i, j+1, k) - 2z(i, j, k) + z(i, j-1, k)}{d^2} - \\ & \mu \frac{z(i, j, k+1) - z(i, j, k-1)}{2t} \end{aligned} \quad (12.24)$$

给定当前位移 $z(i, j, k)$ 和前一个时间段的位移 $z(i, j, k-1)$,要计算出经过时间段 t 后的位移 $z(i, j, k+1)$ 。解方程(12.24)可得 $z(i, j, k+1)$ 的表达式为

$$z(i, j, k+1) = \frac{4 - 8c^2 t^2 / d^2}{\mu t + 2} z(i, j, k) + \frac{\mu t - 2}{\mu t + 2} z(i, j, k-1) + \quad (12.25)$$

$$\frac{2c^2 t^2 / d^2}{\mu t + 2} [z(i+1, j, k) + z(i-1, j, k) + z(i, j+1, k) + z(i, j-1, k)]$$

其中,每一项前的常量可以提前计算出来,则对于网格中的每个顶点只需进行剩余的三个乘法运算和四个加法运算。

若波速 c 太快或者时间间隔 t 太长,则方程(12.25)的连续迭代会导致位移趋向无穷。为了防止位移趋向无穷大,需要确定方程(12.25)保持稳定的准确条件。当位于起伏的流体表面上的一个顶点被释放时,该顶点应该向水平表面移动,这样才可以保证方程(12.25)的收敛。

假设含有 $n \times m$ 个顶点的数组,并且除了坐标为 (i_0, j_0) 的顶点之外,其他顶点均满足 $z(i, j, 0) = 0$ 和 $z(i, j, 1) = 0$ 的条件。令坐标为 (i_0, j_0) 处的顶点经过 $z(i_0, j_0, 0) = h$ 和 $z(i_0, j_0, 1) = h$ 的位置,其中 h 为一个非零的位移。现假设位于 (i_0, j_0) 处的顶点在时间 $2t$ 时被释放,也就是说计算 $z(i_0, j_0, 2)$ 的值时,方程(12.25)的第三项为 0,所以

$$\begin{aligned} z(i_0, j_0, 2) &= \frac{4 - 8c^2t^2/d^2}{\mu t + 2} z(i_0, j_0, 1) + \frac{\mu t - 2}{\mu t + 2} z(i_0, j_0, 0) \\ &= \frac{2 - 8c^2t^2/d^2 + \mu t}{\mu t + 2} h \end{aligned} \quad (12.26)$$

因为顶点向水平表面移动,所以它的位移在 $2t$ 时刻一定比在 t 时刻小,因此下式必须成立:

$$|z(i_0, j_0, 2)| < |z(i_0, j_0, 1)| = |h| \quad (12.27)$$

将方程(12.26)给出的值 $z(i_0, j_0, 2)$ 代入,得

$$\left| \frac{2 - 8c^2t^2/d^2 + \mu t}{\mu t + 2} \right| |h| < |h| \quad (12.28)$$

这样

$$-1 < \frac{2 - 8c^2t^2/d^2 + \mu t}{\mu t + 2} < 1 \quad (12.29)$$

解关于 c 的不等式得

$$0 < c < \frac{d}{2t} \sqrt{\mu t + 2} \quad (12.30)$$

这样,对于相邻顶点之间的距离 d 和方程(12.25)中两个连续迭代之间的时间间隔 t ,波速 c 必须满足方程(12.30)中的最大值,才可保证方程(12.25)的收敛性。

另外,对于给定的相邻顶点之间的间距 d 和波速 c ,也可以计算使方程(12.25)稳定的最大时间间隔 t 。方程(12.29)两边同乘 $-(\mu t + 2)$ 并化简得

$$0 < \frac{4c^2}{d^2} t^2 < \mu t + 2 \quad (12.31)$$

从左边的不等式可以得出 $t > 0$,这是任何情况下都必须满足的条件。右边的不等式可以表示成一个二次不等式

$$\frac{4c^2}{d^2} t^2 - \mu t + 2 < 0 \quad (12.32)$$

相应二次方程的根为

$$t = \frac{\mu \pm \sqrt{\mu^2 + 32c^2/d^2}}{8c^2/d^2} \quad (12.33)$$

由于方程式(12.32)中二次项的系数为正,那么其函数图形为上凹的抛物线,当 t 位于两根之间时,多项式的值为负。因为在方程(12.33)中根式下面的值大于 μ ,所以这两个根中较小根为负值可以被忽略。时间区间 t 的取值范围为

$$0 < t < \frac{\mu + \sqrt{\mu^2 + 32c^2/d^2}}{8c^2/d^2} \quad (12.34)$$

使用方程(12.30)以外的波速值 c ,或者方程(12.34)以外的时间间隔值 t 都会导致顶点位移的指数发散。

12.4 算法实现

为了编程实现方程(12.25)所描述的流体表面方程,需要开设两个缓冲区,并且每一个为包含 $n \times m$ 个顶点位置的数组。在显示每一帧时,其中一个缓冲区包含当前的顶点位置,称为当前缓冲区,另一个缓冲区则包含上一时间段的顶点位置,称为旧缓冲区。在计算顶点的新位移时,需要用新的顶点位移更新旧缓冲区中的顶点位置,则当前缓冲区成为下一帧中的旧缓冲区,而旧缓冲区则成为下一帧中的当前缓冲区。这样,就交替使用这两个缓冲区进行流体表面的显示。

为了减少计算量,需要知道在每一个顶点上的准确法向量和切相量(如果可能)。在坐标为 (i, j) 的顶点处,与 x 轴的平行的切向量 \mathbf{T} 和与 y 轴平行的切向量 \mathbf{B} (这两个向量均为非规范向量),可表示为

$$\begin{aligned}\mathbf{T} &= \left\langle 1, 0, \frac{\partial}{\partial x} z(i, j, k) \right\rangle \\ \mathbf{B} &= \left\langle 0, 1, \frac{\partial}{\partial y} z(i, j, k) \right\rangle\end{aligned}\quad (12.35)$$

将公式(12.15)和公式(12.16)给出的偏导数代入上式,得

$$\begin{aligned}\mathbf{T} &= \left\langle 1, 0, \frac{z(i+1, j, k) - z(i-1, j, k)}{2d} \right\rangle \\ \mathbf{B} &= \left\langle 0, 1, \frac{z(i, j+1, k) - z(i, j-1, k)}{2d} \right\rangle\end{aligned}\quad (12.36)$$

法向量 \mathbf{N} (非规则化向量)可通过计算 $\mathbf{N} = \mathbf{T} \times \mathbf{B}$ 得到,即

$$\begin{aligned}\mathbf{N} &= \begin{vmatrix} i & j & k \\ 1 & 0 & \frac{z(i+1, j, k) - z(i-1, j, k)}{2d} \\ 0 & 1 & \frac{z(i, j+1, k) - z(i, j-1, k)}{2d} \end{vmatrix} \\ &= \left\langle -\frac{z(i+1, j, k) - z(i-1, j, k)}{2d}, -\frac{z(i, j+1, k) - z(i, j-1, k)}{2d}, 1 \right\rangle\end{aligned}\quad (12.37)$$

将向量 \mathbf{T} , \mathbf{B} 和 \mathbf{N} 分别乘以 $2d$ 消去表达式中的分母,这不会改变这些向量的方向,则它们的最终表达式为

$$\begin{aligned}\mathbf{T} &= \langle 2d, 0, z(i+1, j, k) - z(i-1, j, k) \rangle \\ \mathbf{B} &= \langle 0, 2d, z(i, j+1, k) - z(i, j-1, k) \rangle \\ \mathbf{N} &= \langle z(i-1, j, k) - z(i+1, j, k), z(i, j-1, k) - z(i, j+1, k), 2d \rangle\end{aligned}\quad (12.38)$$

程序 12.1 中的代码阐明模拟实现一个流体表面的过程,有一点必须知道,计算流体表面顶点位移时的时间间隔为常数,而大多数游戏的显示帧率变化很大,所以在帧率较高的情况下,必须采用一些方法保证流体表面位置在足够时间后才进行更新。

当一个物体与流体表面接触时(例如,掷一块砖头)会引起流体表面的振动,这时流体表

面的位置可以通过修改接触点周围顶点的位移直接得到。将离接触点最近的点压缩为一点可以获得满意的模拟结果,实践证明将离接触点最近的8个相邻顶点压缩为一点较为合适。

【程序 12.1】 以下程序实现了双缓冲区表面位移算法。类 Fluid 的构造函数含有顶点数组的尺寸、相邻顶点之间的距离 d 、时间间隔 t 、波速 c 以及粘性系数 μ ,其成员变量 renderBuffer 表示当前帧要显示的缓冲区,其取值随着对函数 Fluid::Evaluate 的调用而在 0 和 1 之间变化。

```

struct Vector3D
{
    float      x,y,z;

    Vector3D& Set(float r, float s, float t)
    {
        x = r;
        y = s;
        z = t;
        return (*this);
    }
};

class Fluid
{
private:
    long  width;
    long  height;

    Vector3D *buffer[2];
    Long  renderBuffer;

    Vector3D *normal;
    Vector3D *tangent;

    Float  k1,k2,k3;

public:
    Fluid(long n, long m, float d, float t, float c, float mu);
    ~Fluid();

```

```
        void Evaluate(void);
};

Fluid::Fluid(long n, long m, float d, float t, float c, float mu)
{
    width = n;
    height = m;
    long count = n * m;

    buffer[0] = new Vector3D[count];
    buffer[1] = new Vector3D[count];
    render Buffer=0;

    normal = new Vector3D[count];
    tangent= new Vector3D[count];

    // 计算方程 (12.25) 的常数
    float f1 = c * c * t * t / (d * d);
    float f2 = 1.0F / (mu * t + 2);
    k1 = (4.0F - 8.0F * f1) * f2;
    k2 = (mu * t - 2) * f2;
    k3 = 2.0F * f1 * f2;

    // 初始化缓冲区
    long a = 0;
    for (long j = 0; j<m; j++)
    {
        float y = d * j;
        for (long i = 0; i<n; i++ )
        {
            buffer[0][a].Set(d * i, y, 0.0F);
            buffer[1][a] = buffer[0][a];
            normal[a].Set(0.0F, 0.0F, 2.0F * d);
            tangent[a].Set(2.0F * d, 0.0F, 0.0F);
            a++;
        }
    }
}

Fluid::~~Fluid()
```

```
{
    delete[] tangent;
    delete[] normal;
    delete[] buffer[1];
    delete[] buffer[0];
}

void Fluid::Evaluate(void)
{
    // 使用方程 (12.25)
    for (long j = 1; j < height-1; j++)
    {
        const Vector3D *crnt = buffer[renderBuffer] + j * width;
        Vector3D *prev = buffer[1 - renderBuffer] + j * width;
        for (long i = 1; i < width-1; i++)
        {
            prev[i].z = k * crnt[i].z + k2 * prev[i].z +
                k3 * (crnt[i+1].z + crnt[i-1].z +
                    crnt[i+width].z + crnt[i-width].z);
        }
    }
    // 交换缓冲区
    renderBuffer = 1 - renderBuffer;

    // 计算法线和切线
    for (long j = 1; j < height - 1; j++)
    {
        const Vector3D *next = buffer[renderBuffer] + j * width;
        Vector3D *nrml = normal + j * width;
        Vector3D *tang = tangent + j * width;
        for (long i = 1; i < width-1; i++)
        {
            nrml[i].x = next[i-1].z - next[i+1].z;
            nrml[i].y = next[i-width].z - next[i+width].z;
            tang[i].z = next[i+1].z - next[i-1].z;
        }
    }
}
```

12.5 本章小结

波方程

在粘性阻尼力作用下的流体表面的二维波方程为

$$\frac{\partial^2 z}{\partial t^2} = c^2 \left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right) - \mu \frac{\partial z}{\partial t}$$

其中,常数 c 为波通过介质时的速度, μ 为介质的粘度。

近似导数

函数 $z(x)$ 的一阶导数可近似表示为

$$\frac{d}{dx} z(x) \approx \frac{z(x+d) - z(x-d)}{2d}$$

其中 d 为表示步距的常数。函数 $z(x)$ 的二阶导数为

$$\frac{d^2}{dx^2} z(x) = \frac{z(x+d) - 2z(x) + z(x-d)}{d^2}$$

计算表面位移

经过时间 t 之后流体表面一点位移 $z(i, j, k+1)$ 为

$$z(i, j, k+1) = \frac{4 - 8c^2 t^2 / d^2}{\mu t + 2} z(i, j, k) + \frac{\mu t - 2}{\mu t + 2} z(i, j, k-1) + \frac{2c^2 t^2 / d^2}{\mu t + 2} [z(i+1, j, k) + z(i-1, j, k) + z(i, j+1, k) + z(i, j-1, k)]$$

其中 d 为三角网格中相邻两顶点之间的距离。

数值方法的稳定性

给定时间间隔 t , 波速 c 必须满足

$$0 < c < \frac{d}{2t} \sqrt{\mu t + 2}$$

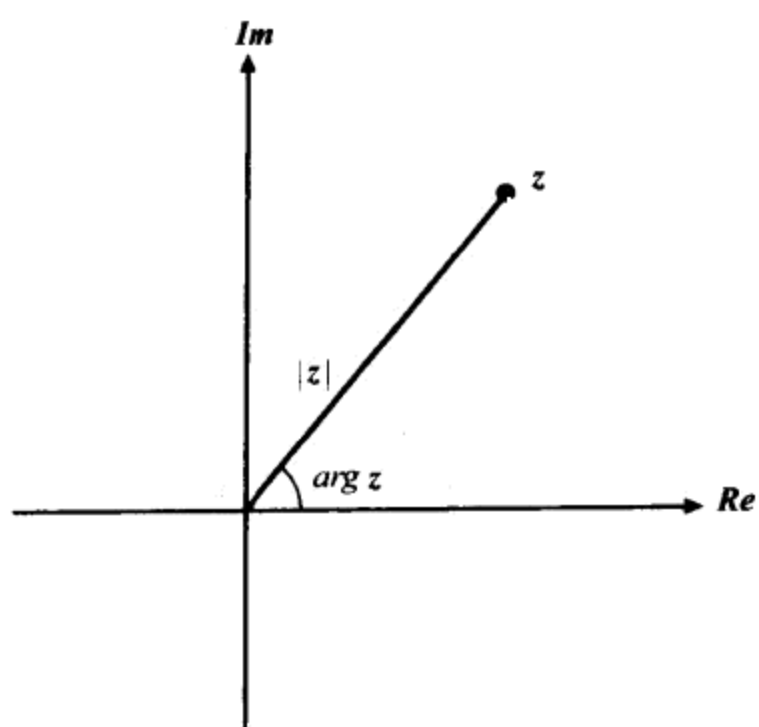
给定波速 c , 时间间隔 t 必须满足

$$0 < t < \frac{\mu + \sqrt{\mu^2 + 32c^2/d^2}}{8c^2/d^2}$$

习 题

1. 假设在三角网格中,每秒计算 20 次每个顶点的表面位移,如果相邻顶点之间的距离 d 为 0.1m,并且阻尼系数为 $\mu = 1s^{-1}$,求要使方程(12.25)收敛的最大波速应为多少?
2. 如习题 1 中所述,在三角网格中相邻两顶点的距离为 0.1m,并且阻尼系数为 $\mu = 1s^{-1}$,求要使波速恒为 2m/s 的两个连续位移之间的最大时间间隔为多少?

附录 A 复数



新
知
屋
PDG

A.1 定义

复数集合 C 包括实数集合 R 以及虚数 i , i 被定义为 -1 的平方根

$$i = \sqrt{-1} \quad (\text{A.1})$$

这样,任意负数 $-n$ 的平方根都可以写为

$$\sqrt{-n} = i\sqrt{n} \quad (\text{A.2})$$

复数 z 的形式是

$$z = a + bi \quad (\text{A.3})$$

这里, a 与 b 都为实数, a 叫做 z 的实部, 记为 $\text{Re}(z)$, b 叫做 z 的虚部, 记为 $\text{Im}(z)$ 。如果 b 是 0, 那么 z 就为实数。如果 $a=0$, 那么 z 就叫做纯虚数。

A.2 加法和乘法

两个复数 $a + bi$ 与 $c + di$ 之和表示为

$$(a + bi) + (c + di) = (a + c) + (b + d)i \quad (\text{A.4})$$

两个复数的乘积可利用乘法的分配律将它们的实部与虚部分别相乘, 同时利用 $i^2 = -1$ 的性质计算出来。复数 $a + bi$ 和 $c + di$ 的积是

$$(a + bi)(c + di) = (ac - db) + (ad + bc)i \quad (\text{A.5})$$

复数的加法运算和乘法运算满足交换律和结合律, 对任意三个复数 z_1, z_2, z_3 , 下面的法则成立, 即

$$(1) z_1 + z_2 = z_2 + z_1$$

$$(2) (z_1 + z_2) + z_3 = z_1 + (z_2 + z_3)$$

$$(3) z_1 z_2 = z_2 z_1$$

$$(4) (z_1 z_2) z_3 = z_1 (z_2 z_3)$$

A.3 共轭复数和复数的逆

一个复数 $z = a + bi$ 的共轭用 \bar{z} 来表示为,

$$\bar{z} = a - bi \quad (\text{A.6})$$

原复数与其共轭复数的实部相同而虚部符号相反, z 和 \bar{z} 的乘积为

$$z\bar{z} = (a + bi)(a - bi) = a^2 + b^2 \quad (\text{A.7})$$

所以, $z\bar{z}$ 的乘积为反映复数大小的实数。用这个乘积表示一个复数的绝对值, 通常叫做复数的模, 复数 $z = a + bi$ 的模就等于

$$|z| = \sqrt{z\bar{z}} = \sqrt{a^2 + b^2} \quad (\text{A.8})$$

如果 z 是纯实数, 它的模就是它的绝对值。

令 $z_1 = a + bi$ 和 $z_2 = c + di$, 且 $z_2 \neq 0$, 那么 z_1 与 z_2 的商 z_1/z_2 可以通过为分子和分母同时乘以 z_2 的共轭复数而计算出来, 即

$$\frac{z_1}{z_2} = \frac{a+bi}{c+di} = \frac{a+bi}{c+di} \cdot \frac{c-di}{c-di} = \frac{(a+bi)(c+di)}{c^2+d^2} = z_1 \frac{\bar{z}_2}{|z_2|^2} \quad (\text{A.9})$$

现在就可以用以下方法来计算一个非零复数 z 的逆 z^{-1}

$$z^{-1} = \frac{\bar{z}}{|z|^2} \quad (\text{A.10})$$

复数 z 和它的逆复数 z^{-1} 的乘积为 1, 即

$$zz^{-1} = \frac{z\bar{z}}{|z|^2} = \frac{z\bar{z}}{z\bar{z}} = 1 \quad (\text{A.11})$$

A.4 DeMoivre 公式

复数的一个突出的优点在于它将指数函数与三角函数联系在一起。对任意表示弧度的实数 x , 式(A.12)成立

$$e^{ix} = \cos x + i \sin x \quad (\text{A.12})$$

这个等式是有名的 DeMoivre 公式, 利用它可以推导出大量的三角恒等式(见 B.4 节)。DeMoivre 公式可通过将函数 e^{ix} 扩展为幂级数并合并实数项和虚数项得到验证(见 D.3 节)。

复平面是二维坐标平面, 有相互垂直的一个实轴和一个虚轴, 如图 A.1 所示。复数 z 可以在此平面上惟一被确定地被表示出来, 其中, 长度是该复数的绝对值, 角度是该复数与复平面的实轴之间的夹角, 这个角叫复角, 记为 $\arg z$, 复数 $z = a + bi$ 的复角的取值范围为

$$\arg z = \begin{cases} \tan^{-1} \frac{a}{b}, & a \geq 0 \\ \tan^{-1} \frac{a}{b} + \pi \operatorname{sgn}(b), & a < 0 \end{cases} \quad (\text{A.13})$$

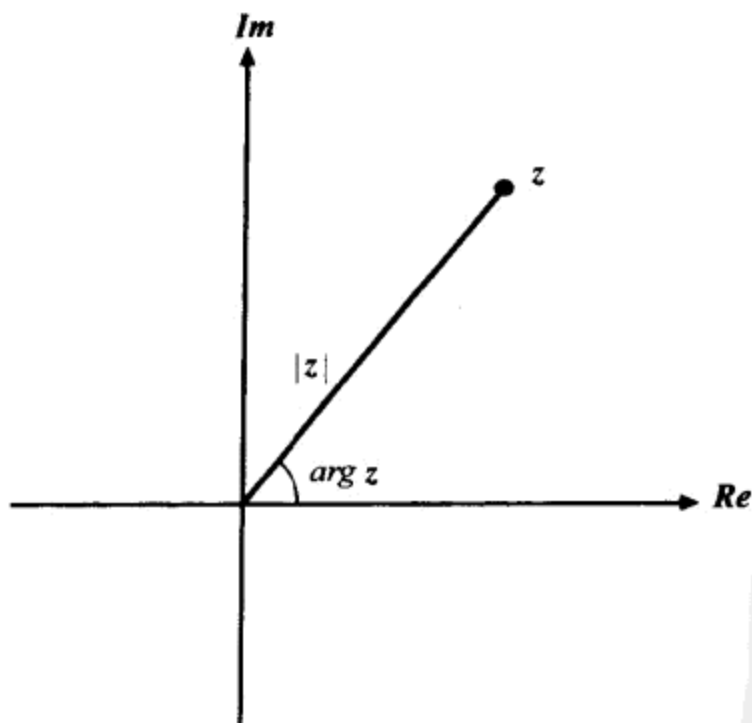


图 A.1 一个复数可以用与原点的距离和与实轴夹角来表示

其中, 函数 $\operatorname{sgn}(b)$ 就是 b 的符号

$$\operatorname{sgn}(b) = \begin{cases} 1, & b \geq 0 \\ -1, & b < 0 \end{cases} \quad (\text{A.14})$$

任何与方程(A.13)中给出的角之间的差为 2π 整数倍的角也是正确的。

现在,对于任意复数 z 可表示为

$$z = re^{i\theta} \quad (\text{A.15})$$

其中 $r = |z|$ 且 $\theta = \operatorname{arg}z$ 。由于 \sin 和 \cos 函数以 2π 为周期,则

$$e^{i\theta} = e^{i(\theta+2k\pi)} \quad (\text{A.16})$$

上式对于任意整数 k 均成立。

德摩根公式可将复数变换为幂的形式, z^n 可以写为

$$z^n = r^n e^{in\theta} = r^n (\cos n\theta + i \sin n\theta) \quad (\text{A.17})$$

特别地,通过式(A.18)可以计算出复数 z 的 n 次方根

$$z^{1/n} = (re^{i(\theta+2\pi k)})^{1/n} = \sqrt[n]{r} \left(\cos \frac{\theta + 2\pi k}{n} + i \sin \frac{\theta + 2\pi k}{n} \right) \quad (\text{A.18})$$

其中, k 是整数,令 $k = 0, 1, \dots, n-1$ 可以得到复数 z 的所有 n 个根。对于根 ρ , 如果其最小整数幂 $\rho^m = z$ 时 $m = n$, 则 ρ 叫做基本根。

当 $r = 1$ 和 $\theta = 0$ 时,复数为单位复数,它的 n 次方根可以表示为

$$e^{2\pi ki/n} = \cos \frac{2\pi k}{n} + i \sin \frac{2\pi k}{n} \quad (\text{A.19})$$

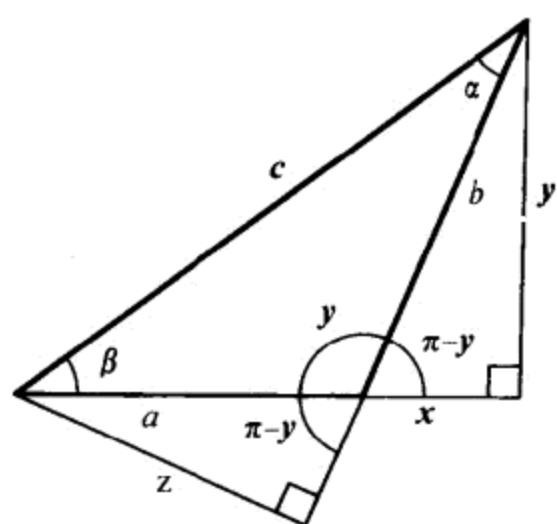
例如,单位复数的 3 个立方根 ρ_1, ρ_2 和 ρ_3 分别为

$$\begin{aligned} \rho_0 &= 1 \\ \rho_1 &= \cos \frac{2\pi}{3} + i \sin \frac{2\pi}{3} = -\frac{1}{2} + i \frac{\sqrt{3}}{2} \\ \rho_2 &= \cos \frac{4\pi}{3} + i \sin \frac{4\pi}{3} = -\frac{1}{2} - i \frac{\sqrt{3}}{2} \end{aligned} \quad (\text{A.20})$$

其中, ρ_1 与 ρ_2 都是单位复数的基本根,且 $\rho_1^2 = \rho_2$ 且 $\rho_2^2 = \rho_1$ 。通常,一个单位复数的基本 n 次方根可以产生单位复数的所有 n 次根,这时只要依次取它的 $1, 2, \dots, n$ 次幂。



附录 B 三角恒等式



B.1 三角函数

对于图 B.1 中所示的三角形的角 α , 其三角函数的定义如下:

$$\begin{aligned} \sin\alpha &= \frac{y}{r} & \cos\alpha &= \frac{x}{r} \\ \tan\alpha &= \frac{y}{x} & \cot\alpha &= \frac{x}{y} \\ \sec\alpha &= \frac{r}{x} & \csc\alpha &= \frac{r}{y} \end{aligned} \quad (\text{B.1})$$

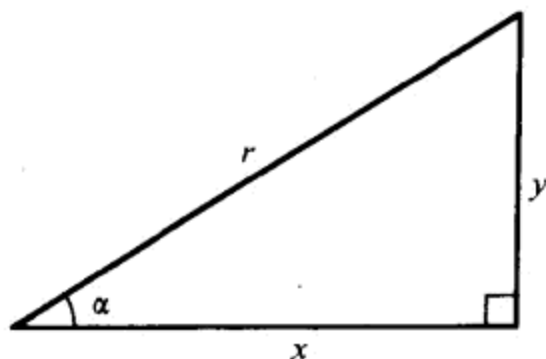


图 B.1 以三角形的 α 角为参数定义三角函数

这些三角函数的关系如下:

$$\begin{aligned} \tan\alpha &= \frac{\sin\alpha}{\cos\alpha} & \cot\alpha &= \frac{1}{\tan\alpha} \\ \sec\alpha &= \frac{1}{\cos\alpha} & \csc\alpha &= \frac{1}{\sin\alpha} \end{aligned} \quad (\text{B.2})$$

B.2 对称与移相

余弦函数 \cos 是偶函数, 即它是关于 y 轴对称的, 正弦函数 \sin 和正切函数 \tan 都是奇函数, 它们关于原点对称, 则

$$\begin{aligned} \sin(-\alpha) &= -\sin\alpha \\ \cos(-\alpha) &= \cos\alpha \\ \tan(-\alpha) &= -\tan\alpha \end{aligned} \quad (\text{B.3})$$

余弦函数在其角度为 α 时的值等于正弦函数角度为 $\alpha + \pi/2$ 时的值。即余弦函数的图形等于正弦函数的图形向左平移 $\pi/2$ 后的图形。则移相恒等式为

$$\begin{aligned} \sin(\alpha + \pi/2) &= \cos\alpha \\ \cos(\alpha + \pi/2) &= -\sin\alpha \\ \tan(\alpha + \pi/2) &= -\cot\alpha \end{aligned} \quad (\text{B.4})$$

再结合(B.3)给出的对称恒等式, 还可以得到下面这组恒等式

$$\begin{aligned} \sin(\pi/2 - \alpha) &= \cos\alpha \\ \cos(\pi/2 - \alpha) &= \sin\alpha \end{aligned}$$

$$\tan(\pi/2 - \alpha) = \cot\alpha \quad (\text{B.5})$$

将正弦函数和余弦函数向左平移角度 π 后的函数值只是原函数值的取反,即

$$\begin{aligned} \sin(\alpha + \pi) &= -\sin\alpha \\ \cos(\alpha + \pi) &= -\cos\alpha \\ \tan(\alpha + \pi) &= \tan\alpha \end{aligned} \quad (\text{B.6})$$

再一次利用三角函数的对称恒等式,可得到

$$\begin{aligned} \sin(\pi - \alpha) &= \sin\alpha \\ \cos(\pi - \alpha) &= -\cos\alpha \\ \tan(\pi - \alpha) &= -\tan\alpha \end{aligned} \quad (\text{B.7})$$

B.3 毕达哥拉斯恒等式

下面的一些恒等式是根据勾股定理 $x^2 + y^2 = r^2$, 直接从方程(B.1)中得出的

$$\begin{aligned} \sin^2\alpha + \cos^2\alpha &= 1 \\ \tan^2\alpha + 1 &= \sec^2\alpha \\ \cot^2\alpha + 1 &= \csc^2\alpha \end{aligned} \quad (\text{B.8})$$

其中,如果角 α 的范围为 $0 \leq \alpha \leq \pi/2$, 则以下等式成立:

$$\begin{aligned} \sin\alpha &= \sqrt{1 - \cos^2\alpha} \\ \cos\alpha &= \sqrt{1 - \sin^2\alpha} \\ \tan\alpha &= \sqrt{\sec^2\alpha - 1} \end{aligned} \quad (\text{B.9})$$

B.4 指数恒等式

DeMoivre 公式为

$$e^{ai} = \cos\alpha + i\sin\alpha \quad (\text{B.10})$$

利用与指数相关的定律,从 DeMoivre 公式能够推导出多个三角恒等式。角的和与差恒等式可通过下式推导出来:

$$e^{(a+\beta)i} = e^{ai}e^{\beta i} \quad (\text{B.11})$$

再用方程(B.10)扩展式(B.11)得

$$\cos(\alpha + \beta) + i\sin(\alpha + \beta) = (\cos\alpha + i\sin\alpha)(\cos\beta + i\sin\beta) \quad (\text{B.12})$$

令方程两边的实部和虚部分别相等,可得

$$\begin{aligned} \sin(\alpha + \beta) &= \sin\alpha\cos\beta + \cos\alpha\sin\beta \\ \cos(\alpha + \beta) &= \cos\alpha\cos\beta - \sin\alpha\sin\beta \end{aligned} \quad (\text{B.13})$$

两个角之差的公式为

$$\begin{aligned} \sin(\alpha - \beta) &= \sin\alpha\cos\beta - \cos\alpha\sin\beta \\ \cos(\alpha - \beta) &= \cos\alpha\cos\beta + \sin\alpha\sin\beta \end{aligned} \quad (\text{B.14})$$

当 α 与 β 相等时,它们之和的三角函数为

$$\sin 2\alpha = 2\sin\alpha\cos\alpha$$

$$\cos 2\alpha = \cos^2 \alpha - \sin^2 \alpha \quad (\text{B.15})$$

利用 $\sin^2 \alpha + \cos^2 \alpha = 1$ 的事实, 可以将 $\cos 2\alpha$ 再度表示为

$$\begin{aligned} \cos 2\alpha &= 1 - 2\sin^2 \alpha \\ \cos 2\alpha &= 2\cos^2 \alpha - 1 \end{aligned} \quad (\text{B.16})$$

从上式中解出 $\sin^2 \alpha$ 与 $\cos^2 \alpha$, 得

$$\begin{aligned} \sin^2 \alpha &= \frac{1 - \cos 2\alpha}{2} \\ \cos^2 \alpha &= \frac{1 + \cos 2\alpha}{2} \end{aligned} \quad (\text{B.17})$$

B.5 反三角函数

三角函数 $f(\alpha)$ 的反函数 $f^{-1}(x)$ 返回 $f(\alpha) = x$ 中的角 α , 下表中列出了反三角函数的定义域和值域。

函 数	定 义 域	值 域
$\sin^{-1} x$	$[-1, 1]$	$[-\pi/2, \pi/2]$
$\cos^{-1} x$	$[-1, 1]$	$[0, \pi]$
$\tan^{-1} x$	R	$[-\pi/2, \pi]$

如图 B.2 所示, x 的反正弦函数等于三角形中的锐角 α , 该角在图中对应边长为 x , 且斜边长为 1 的角。既然已知第三边长为 $\sqrt{1-x^2}$, 则可以导出角 $\sin^{-1} x$ 的其他三角函数

$$\begin{aligned} \cos(\sin^{-1} x) &= \sqrt{1-x^2} \\ \tan(\sin^{-1} x) &= \frac{x}{\sqrt{1-x^2}} \end{aligned} \quad (\text{B.18})$$

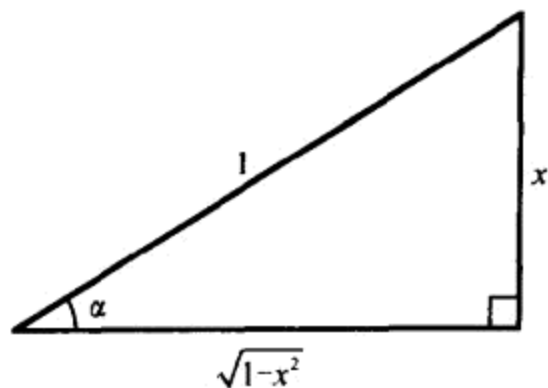


图 B.2 表示正弦函数的反三角函数的直角三角形

运用同样的方法可以导出余弦函数 \cos 与正切函数 \tan 的反函数

$$\begin{aligned} \sin(\cos^{-1} x) &= \sqrt{1-x^2} \\ \tan(\cos^{-1} x) &= \frac{\sqrt{1-x^2}}{x} \end{aligned}$$



$$\begin{aligned}\sin(\tan^{-1}x) &= \frac{x}{\sqrt{x^2+1}} \\ \cos(\tan^{-1}x) &= \frac{1}{\sqrt{x^2+1}}\end{aligned}\quad (\text{B.19})$$

B.6 正弦与余弦法则

观察图 B.3 中的三角形,可以得出以下结论,即

$$\begin{aligned}\sin\alpha &= \frac{z}{c} \\ \sin\beta &= \frac{y}{c}\end{aligned}\quad (\text{B.20})$$

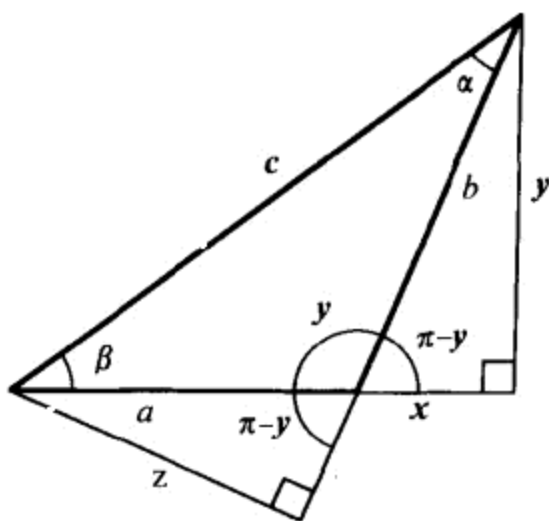


图 B.3 说明正弦与余弦法则的一个三角形

解关于 c 的方程,得

$$\frac{z}{\sin\alpha} = \frac{y}{\sin\beta}\quad (\text{B.21})$$

由图 B.3 可得出以下方程:

$$\begin{aligned}\sin(\pi - r) &= \frac{z}{a} \\ \sin(\pi - r) &= \frac{y}{b}\end{aligned}\quad (\text{B.22})$$

所以 $z/a = y/b$ 。在方程(B.21)左端乘以 z/a ,右端乘以 y/b ,即得余弦法则为

$$\frac{a}{\sin\alpha} = \frac{b}{\sin\beta}\quad (\text{B.23})$$

角 α 和角 γ 也满足同样的关系式,则

$$\frac{a}{\sin\alpha} = \frac{b}{\sin\beta} = \frac{c}{\sin\gamma}\quad (\text{B.24})$$

现在,分析图 B.3 所示直角三角形的毕达哥拉斯恒等式

$$\begin{aligned}x^2 + y^2 &= b^2 \\ (a+x)^2 + y^2 &= c^2\end{aligned}\quad (\text{B.25})$$

从第一个方程中解出 y^2 , 代入第二个方程, 得

$$c^2 = (a + x)^2 + b^2 - x^2 = a^2 + b^2 + 2ax \quad (\text{B.26})$$

x 的值可由以下方法代替, 即

$$\cos(\pi - r) = \frac{x}{b} \quad (\text{B.27})$$

由于 $\cos(\pi - r) = -\cos\gamma$, 所以

$$x = -b\cos\gamma \quad (\text{B.28})$$

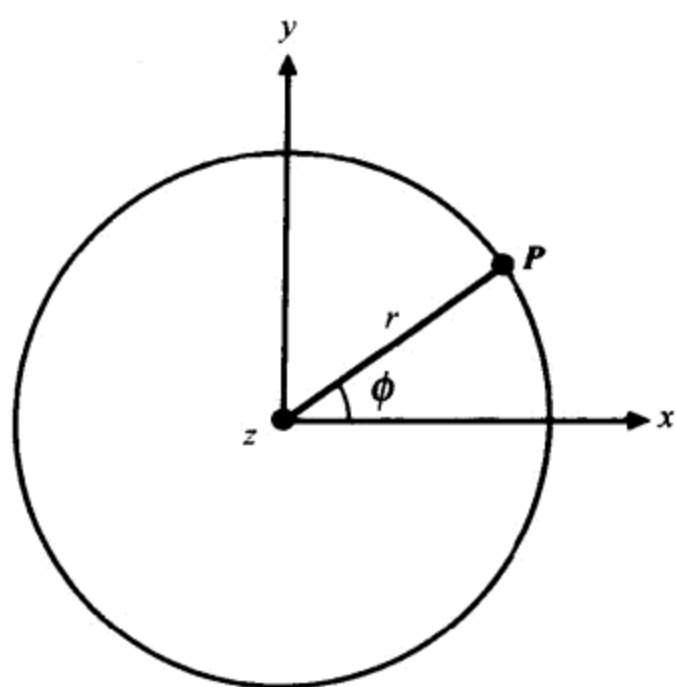
把它代入方程(B.26), 则得余弦法则

$$c^2 = a^2 + b^2 - 2ab\cos\gamma \quad (\text{B.29})$$

当然, 如果 γ 是一个直角, 则由于 $\cos\pi/2 = 0$, 余弦法则又变为毕达哥拉斯定理。



附录 C 坐标系



C.1 圆柱坐标系

如图 C.1 所示,圆柱坐标系中的一点 P 可以用 r, θ 以及 z 表示。 r 等于 P 和 z 轴之间的径向距离, θ 等于连接 P 在 xy 平面上的投影和原点的直线与 x 轴之间的逆时针方向的夹角。

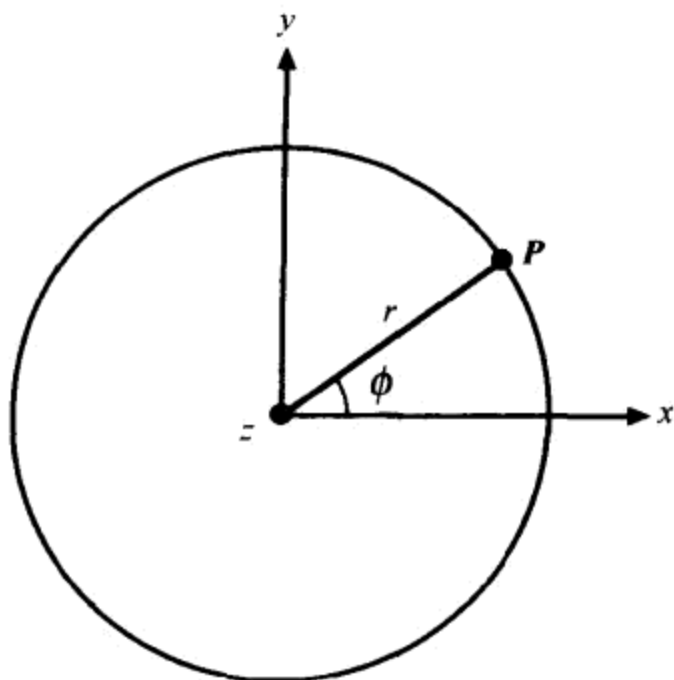


图 C.1 圆柱坐标系, z 轴垂直指出页面

圆柱坐标系中的 z 坐标与在笛卡尔坐标系中的 z 坐标相同,圆柱坐标为 $\langle r, \theta, z \rangle$ 的点在笛卡尔坐标系中的 x, y 可表示为

$$\begin{aligned} x &= r \cos \theta \\ y &= r \sin \theta \end{aligned} \quad (\text{C.1})$$

如果笛卡尔坐标系中 x 与 y 已知,那么对应的圆柱系中坐标 r 与 θ 可表示为

$$\begin{aligned} r^2 &= \sqrt{x^2 + y^2} \\ \theta &= \text{sgn}(y) \cos^{-1} \frac{x}{r} \end{aligned} \quad (\text{C.2})$$

其中,函数 $\text{sgn}(y)$ 为符号函数

$$\text{sgn}(y) = \begin{cases} 1, & y \geq 0 \\ -1, & y < 0 \end{cases} \quad (\text{C.3})$$

角 θ 也可以表示为

$$\theta = \begin{cases} \tan^{-1} \frac{y}{x}, & x \geq 0 \\ \tan^{-1} \frac{y}{x} + \pi \text{sgn}(y), & x < 0 \end{cases} \quad (\text{C.4})$$

在方程(C.2)和(C.4)中, θ 的值满足 $-\pi \leq \theta \leq \pi$ 。

C.2 球坐标系

如图 C.2 所示,球面坐标系中一点 P 可以用 r, θ 和 φ 表示。 r 等于 P 到坐标原点的距离, θ 叫极角,它是原点与 P 所连直线与 z 轴之间的夹角,极角 θ 总满足条件 $0 \leq \theta \leq \pi$,角 φ 叫方位角,或者简称为方位,它代表连接 P 在 xy 平面上的投影和原点的直线与 x 轴之间的逆时针方向的夹角(与圆柱坐标系中相同)。

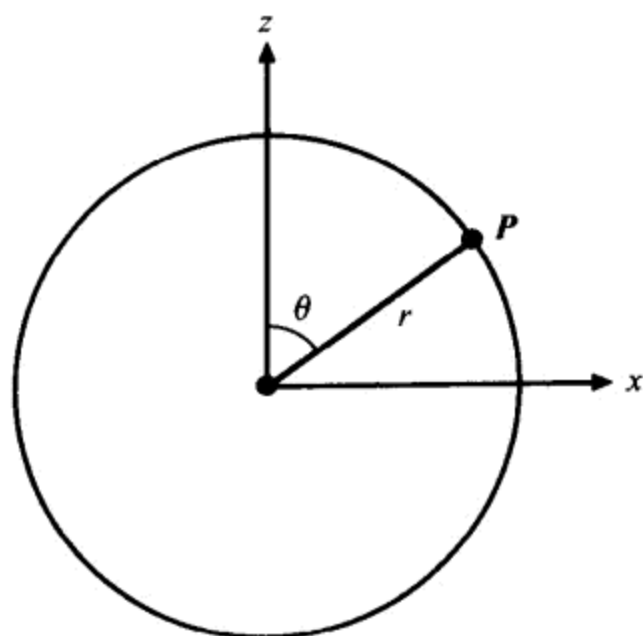


图 C.2 球坐标系

笛卡尔坐标系中 $\langle x, y, z \rangle$ 与球坐标系中 $\langle r, \theta, \varphi \rangle$ 之间的对应关系如下,即

$$\begin{aligned} x &= r \sin \theta \cos \varphi \\ y &= r \sin \theta \sin \varphi \\ z &= r \cos \theta \end{aligned} \quad (\text{C.5})$$

用以下关系式可以将笛卡尔坐标变换为球坐标

$$\begin{aligned} r &= \sqrt{x^2 + y^2 + z^2} \\ \theta &= \cos^{-1} \frac{z}{r} \\ \varphi &= \text{sgn}(y) \cos^{-1} \frac{x}{\sqrt{x^2 + y^2}} \end{aligned} \quad (\text{C.6})$$

其中,方位角 φ 也可以利用方程(C.4)计算出来。



附录 D 泰勒级数

$$\begin{aligned} f(x) &= f(c) + f'(c)(x - c) + \frac{f''(c)}{2!}(x - c)^2 + \frac{f'''(c)}{3!}(x - c)^3 + \dots \\ &= \sum_{k=0}^{\infty} \frac{f^{(k)}(c)}{k!}(x - c)^k \end{aligned}$$



D.1 导数

令函数 $f(x)$ 在区间 I 上存在直到 n 阶的导数, 假设通过以下 n 次多项式 $p_n(x)$ 近似表示区间 I 中 $x=c$ 处的函数 $f(x)$ 的值, 则

$$p_n(x) = a_0 + a_1(x-c) + a_2(x-c)^2 + \cdots + a_n(x-c)^n = f(x) \quad (\text{D.1})$$

这里 x 可以为任意小的数。 $p_n(x)$ 在 $x=c$ 处的导数值为

$$\begin{aligned} p_n'(c) &= a_1 \\ p_n''(c) &= 2a_2 \\ &\vdots \\ p_n^{(n)}(c) &= n!a_n \end{aligned} \quad (\text{D.2})$$

通过以下条件可以确定系数, 即

$$\begin{aligned} p_n(c) &= f(c) \\ p_n'(c) &= f'(c) \\ p_n''(c) &= f''(c) \\ &\vdots \\ p_n^{(n)}(c) &= f^{(n)}(c) \end{aligned} \quad (\text{D.3})$$

这样, 多项式 $P_n(x)$ 就可表示为

$$p_n(x) = f(x) + f'(c)(x-c) + \frac{f''(c)}{2!}(x-c)^2 + \cdots + \frac{f^{(n)}(c)}{n!}(x-c)^n \quad (\text{D.4})$$

令误差项 $r_n(x)$ 为函数 $f(x)$ 的准确值与近似值 $P_n(x)$ 之间的差, 则

$$\begin{aligned} f(x) &= f(c) + f'(c)(x-c) + \frac{f''(c)}{2!}(x-c)^2 + \cdots + \\ &\quad \frac{f^{(n)}(c)}{n!}(x-c)^n + r_n(x) \end{aligned} \quad (\text{D.5})$$

令函数 $g(x)$ 为

$$\begin{aligned} g(z) &= f(x) - f(z) - f'(z)(x-z) - \frac{f''(z)}{2!}(x-z)^2 - \cdots - \\ &\quad \frac{f^{(n)}(z)}{n!}(x-z)^n - \frac{(x-z)^{n+1}}{(x-c)^{n+1}}r_n(x) \end{aligned} \quad (\text{D.6})$$

用(D.5)方程很容易证明 $g(x)=0$ 并且 $g(c)=0$ 。 $g(z)$ 的导数可简单表示为

$$g'(z) = -\frac{f^{(n+1)}(z)}{n!}(x-z)^n + (n+1)\frac{(x-z)^n}{(x-z)^{n+1}}r_n(x) \quad (\text{D.7})$$

根据 Rolle 定理, 在 x 与 c 之间存在 z_0 使 $g'(z_0)=0$ 。 计算出 $g'(z_0)$ 并解关于 $r_n(x)$ 的方程得

$$r_n(x) = \frac{f^{(n+1)}(z_0)}{(n+1)!}(x-c)^{n+1} \quad (\text{D.8})$$

如果函数 $f(x)$ 是无限可微分的函数, 那么

$$\lim_{n \rightarrow \infty} r_n(x) = 0 \quad (\text{D.9})$$

则函数 $f(x)$ 表示为下面的无穷级数, 即

$$\begin{aligned} f(x) &= f(c) + f'(c)(x-c) + \frac{f''(c)}{2!}(x-c)^2 + \frac{f'''(c)}{3!}(x-c)^3 + \cdots \\ &= \sum_{k=0}^{\infty} \frac{f^{(k)}(c)}{k!}(x-c)^k \end{aligned} \quad (\text{D.10})$$

D.2 幂级数

令方程(D.10)中 $c=0$, 可推导出一般函数的幂级数展开式。因为指数函数 e^x 等于它自己的导数, 且 $e^0=1$, 则它的幂级数为

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots = \sum_{k=0}^{\infty} \frac{x^k}{k!} \quad (\text{D.11})$$

对于正弦函数, 有

$$\begin{aligned} f(x) &= \sin x & f(0) &= 0 \\ f'(x) &= \cos x & f'(0) &= 1 \\ f''(x) &= -\sin x & f''(0) &= 0 \\ f'''(x) &= -\cos x & f'''(0) &= -1 \end{aligned} \quad (\text{D.12})$$

正弦函数的幂级数为

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{(2k+1)!} \quad (\text{D.13})$$

简单而言, 余弦函数的幂级数为

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k}}{(2k)!} \quad (\text{D.14})$$

以下函数也是常用的函数

$$f(x) = \frac{1}{x+1} \quad (\text{D.15})$$

因为它是函数 $\ln(1+x)$ 在区间 $(0, \infty)$ 上的导数, 函数 $f(x)$ 的前几阶导数是

$$\begin{aligned} f'(x) &= \frac{1}{(1+x)^2} \\ f''(x) &= \frac{2}{(1+x)^3} \\ f'''(x) &= \frac{-6}{(1+x)^4} \end{aligned} \quad (\text{D.16})$$

通常, $f(x) = \frac{1}{x+1}$ 的 k 阶导数为

$$f^{(k)}(x) = \frac{(-1)^k k!}{(1+x)^{k+1}} \quad (\text{D.17})$$

令 $x=0$, 得 $f^{(k)}(0) = (-1)^k k!$, 所以 $f(x)$ 的幂级数是

$$\frac{1}{1+x} = 1 - x + x^2 - x^3 + \cdots = \sum_{k=0}^{\infty} (-1)^k x^k \quad (\text{D.18})$$

该级数在区间 $(-1, 1)$ 上收敛, 对(D.18)式两边同求积分, 可得以下的幂级数, 它是在同一区间上 $1+x$ 的自然对数的幂级数

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots = \sum_{k=0}^{\infty} \frac{(-1)^k x^{k+1}}{k+1} \quad (\text{D.19})$$

D.3 DeMoivre 等式

DeMoivre 等式表示了指数函数、正弦函数以及余弦函数之间的关系,即

$$e^{ix} = \cos x + i \sin x \quad (\text{D.20})$$

通过分析函数 e^{ix} 的幂级数可以验证以上关系

$$e^{ix} = \sum_{k=0}^{\infty} \frac{i^k x^k}{k!} \quad (\text{D.21})$$

利用 $i^2 = -1$, $i^3 = -i$ 和 $i^4 = 1$ 的事实,可以把指数函数的实部与虚部的幂级数分别写出来

$$e^{ix} = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k}}{(2k)!} + i \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{(2k+1)!} \quad (\text{D.22})$$

与方程(D.13)和方程(D.14)比较就可得到式(D.20)中的结果。



附录 E 习题答案

$$\begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f}{f-n} & -\frac{nf}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$



第 1 章

1. (1) -2 (2) $\langle 2, 1, -6 \rangle$ (3) $\langle -\frac{4}{9}, -\frac{4}{9}, -\frac{2}{9} \rangle$

2. $\mathbf{e}'_1 = \mathbf{e}_1$, $\mathbf{e}'_2 = \mathbf{e}_2$, $\mathbf{e}'_3 = \langle 1, -1, -2 \rangle$

第 2 章

1. (1) 22 (2) -1 (3) 1 (4) 0

2. (1)
$$\begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{4} \end{bmatrix}$$

(2)
$$\begin{bmatrix} 1 & 0 & 0 \\ \frac{3}{8} & \frac{1}{2} & -\frac{1}{8} \\ -\frac{3}{8} & 0 & \frac{1}{8} \end{bmatrix}$$

(3)
$$\begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

(4)
$$\begin{bmatrix} 1 & 0 & 0 & -4 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & -7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3.
$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = a \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

4. $\lambda_1 = -1, \lambda_2 = 2, \lambda_3 = 5$

第 3 章

1. $\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ 0 & \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}$, $\mathbf{R}_y = \begin{bmatrix} \frac{\sqrt{3}}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \\ -\frac{1}{2} & 0 & \frac{\sqrt{3}}{2} \end{bmatrix}$, $\mathbf{R}_z = \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$

2. $\mathbf{q} = \pm \left[\frac{\sqrt{3}}{2} + \langle 0, \frac{3}{10}, \frac{2}{5} \rangle \right]$

第 4 章

1. $\langle 2, 1, 0, -4 \rangle$ 及其与任意比例系数的乘积

2. 63.1 度

3. 左: $\langle \frac{\sqrt{3}}{2}, 0, -\frac{\sqrt{2}}{2}, 0 \rangle$; 右: $\langle -\frac{\sqrt{2}}{2}, 0, -\frac{\sqrt{2}}{2}, 0 \rangle$;

底: $\langle 0, \frac{4}{5}, -\frac{3}{5}, 0 \rangle$; 顶: $\langle 0, -\frac{4}{5}, -\frac{3}{5}, 0 \rangle$

$$4. \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f}{f-n} & -\frac{nf}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

第 5 章

2. 5.3271783

3. $x_{n+1} = \frac{1}{p}x_n(p+1-rx^p)$

4. $\left(V_x^2 + V_y^2 - \frac{r^2}{h^2}V_z^2\right)t^2 + 2\left[Q_xV_x + Q_yV_y + \frac{r}{h}V_z\left(r - \frac{r}{h}Q_z\right)\right]t + Q_x^2 + Q_y^2 + \frac{r}{h}Q_z(2r - \frac{r}{h}Q_z) - r^2 = 0$

5. $\langle -0.212, 0.636, -0.742 \rangle$

7. 49度

第 6 章

1. 3.16m

2. 4.35m

第 7 章

1. 如果两球体互不包含, $r = \frac{1}{2}(d + r_1 + r_2)$ 和 $Q = Q_1 + \frac{r-r_1}{d}(Q_2 - Q_1)$, 其中,
 $d = \|Q_2 - Q_1\|$ 2. 如果 $s < h$, $r = \frac{s^2 + h^2}{2h}$ 且 $Q = \langle 0, 0, h - r \rangle$
如果 $s \geq h$, $r = s$ 且 $Q = \langle 0, 0, 0 \rangle$

3. $r_{\text{eff}} = \frac{5\sqrt{3}}{6} \approx 1.443$

第 8 章

1. $t = 4s$

第 9 章

1. 1.001

2. 0.003

第 10 章

1. $x(t) = Ae^{3t} + Bte^{3t} + t + 1$

2. $x(t) = 3\cos 4t + \sin 4t$

3. 30.2m

4. 3.93s

5. $t \approx 6.1\text{s}$

6. $a = \frac{g}{M+m} [M - m(\sin\theta + \mu_K \cos\theta)]$

第 11 章

1. $F = m\omega \sqrt{\omega^2 r^2 + 4v^2}$

2. $\omega = \sqrt{\frac{\mu_s g}{r}}$

3. $C = \left\langle 0, 0, \frac{5}{9} \right\rangle$

4. $I = \frac{1}{2} m (R_1^2 + R_2^2)$

5.
$$J = \begin{bmatrix} \frac{1}{12} m (b^2 + c^2) & 0 & 0 \\ 0 & \frac{1}{12} m (a^2 + c^2) & 0 \\ 0 & 0 & \frac{1}{12} m (a^2 + b^2) \end{bmatrix}$$

6. $a = \frac{g}{1 + 1/mR^2} = \frac{g}{1 + M/2m}$

7. $a = \frac{g \sin\theta}{1 + 1/mR^2} = \frac{5}{7} g \sin\theta$

8. $z = \frac{d}{2\mu_s}$

第 12 章

1. 1.43m/s

2. 0.0357s



参考文献

1. Beckmann, Petr, and Spizzichino, André. The Scattering of Electromagnetic Waves from Rough Surfaces. Macmillan, 1963
2. Blinn, Jim. Jim Blinn's Corner: A Trip Down the Graphics Pipeline. Morgan – Kaufmann, 1996
3. Blinn, Jim. Jim Blinn's Corner: Dirty Pixels. Morgan – Kaufmann, 1998
4. Boyce, William E, and DiPrima, Richard C. Elementary Differential Equations and Boundary Value Problems. 4th edition. Wiley, 1986
5. Cook, Robert L, and Torrance, Kenneth E. A Reflectance Model for Computer Graphics. ACM Translations on Graphics, 1982,1(1): 7~24
6. Dummit, David S, and Foote, Richard M. Abstract Algebra. Prentice – Hall, 1991
7. Foley, James D, et al. Computer Graphics: Principles and Practice. 2nd edition. Addison – Wesley, 1990
8. Gomez, Miguel. Interactive Simulation of Water surfaces. Game Programming Gems. Charles River Media, 2000
9. Gonzalez, Rafael C, and Woods, Richard E. Digital Image Processing. Addison – Wesley, 1992
10. MÖller, Tomas, and Haines, Eric. real – time Rendering. AK: Peters, 1999
11. He, Xiao D, et al. A Comprehensive Physical Model for Light Reflection. SIGGRAPH '91, Computer Graphics. 1991,25(4):175~186
12. Johnson, Richard A, and Wichern, Dean W. Applied Multivariate Statistical Analysis. 4th edition. Prentice – Hall, 1998
13. Kautz, Jan, et al. Achieving Real – Time Realistic Refections, Part 1. Game Developer. 2001,8(1): 32~37
14. Kautz, Jan, et al. Achieving Real – Time Realistic Refections. Part 2. Game Developer. 2001,8(2):38~44
15. Lander, Jeff. The Trials and Tribulations of Tribology. Game Developer. 1999,6(8): 9~24
16. Lengyel, Eric. Tweaking A Vertex's Projected Depth Value. Game Programming Gems, Charles River Media, 2000
17. Lengyel, Eric. A Fast Cylinder – Frustum Intersection Test. Game Programming Gems, Charles River Media, 2000
18. Lengyel, Eric. Applying Decals to Arbitrary Surface. Game Programming Gems, Charles River Media, 2000
19. Marion, Jerry B, and Thornton, Stephen T. Classical Dynamics of Particals and systems.

- 3rd edition. Harcourt Brace Jovanovich, 1988
20. O'Neil, Peter, V. Advanced Engineering Mathematics. 3rd edition. PWS, 1991
 21. OpenGL Architecture Review Board. OpenGL Programming Guide. 3rd edition. Addison - Wesley, 1999
 22. OpenGL Architecture Review Board. OpenGL Reference Manual. 3rd edition. Addison - Wesley, 1999
 23. Press, William H, et al. Numerical Recipes in C. Cambridge, 1988
 24. Reitz, John R, Milford, Frederick J, and Christy, Robert W. Foundation of Electromagnetic theory. 4th edition. Addison - Wesley, 1993
 25. Sears, Francis W, Zemansky, Mark W, and Young, Hugh D. University Physics. 7th edition. Addison - Wesley. 1987
 26. Sillion, Francois X, and Peuch, Claude. Radiosity and Global Illumination. Morgan Kaufmann, 1994
 27. Schlick, Christophe. An Inexpensive BRDF Model for Physically - based Rendering. Proc. Eurographics '94. Computer Graphics Forum. 13(3): 233~246
 28. Schwarze, Jochen. Cubic and Quartic Roots. Graphics Gems. Academic Press, 1990
 29. Ward, Gregory J. Measuring and Modeling Anisotropic Reflection. SIGGRAPH '92. Computer Graphics, 1992, 26(2): 265~272
 30. Wu, Xiaolin. A Linear - Time Simple Bounding Volume Algorithm. Graphics Gems III. Academic Press, 1992

