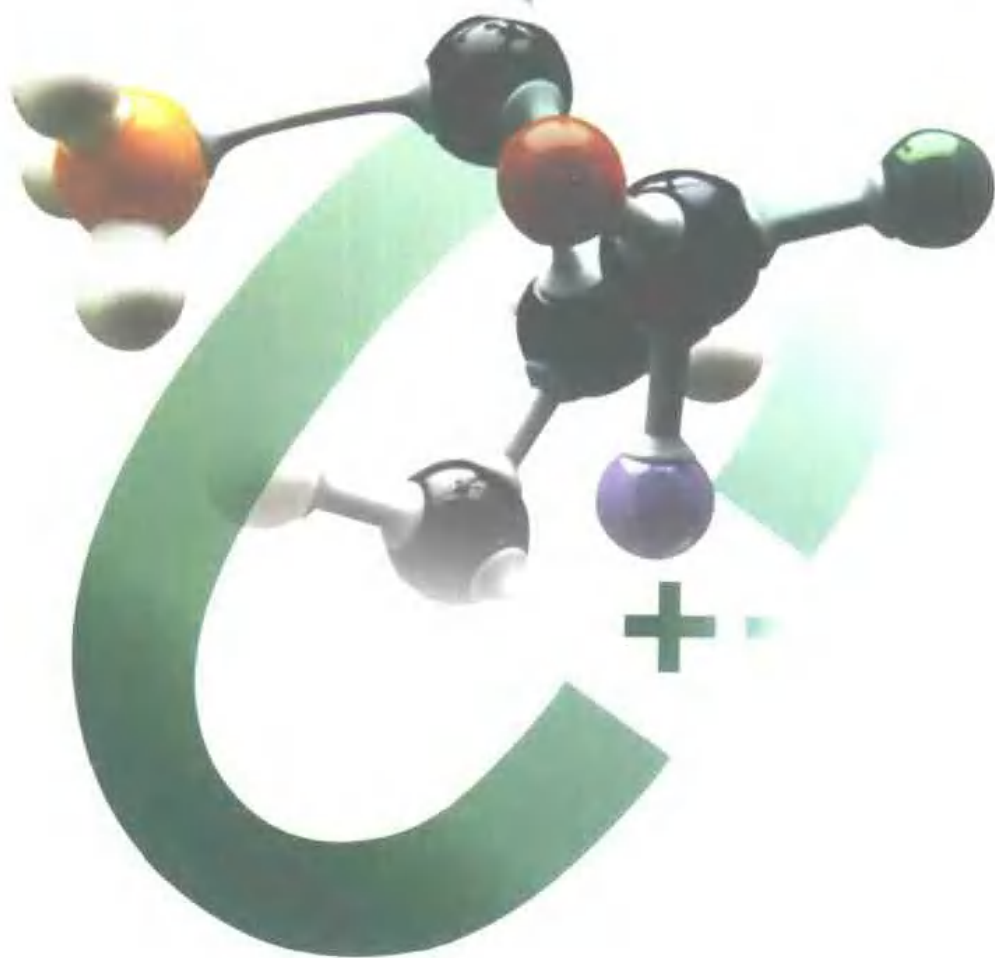




Direct3D原理

与API参考 (上)

武永康 编著



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



Direct3D 原理与 API 参考

(上)

武永康 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

微软的 DirectX 7.0 SDK(软件开发工具包)提供了一套最新的、优秀的应用程序设计接口。这个软件接口可以提供开发高性能、实时的应用程序所需要的各种资源。DirectX 支持二维和三维图形、声音和音乐、输入、力反馈以及多用户游戏应用程序中的网络通信。DirectX 技术有助于创建 Windows 环境的高级电脑游戏和实时多媒体应用程序。Direct3D 是 DirectX 应用程序设计接口中最重要的组件。现在为 Windows 环境提供的 3D 游戏都是用 Direct3D 实现的。本书完整地介绍了 Direct3D,包括 Direct3D 的结构和功能,以及 Direct3D 应用程序设计原理,给出实用的 Direct3D 应用程序设计接口参考信息,以 Visual C++ 和 Visual Basic 两种语言环境为背景,分别用实例阐明了 Direct3D 应用程序设计细节。

本书是 Direct3D 应用程序设计参考书,适用于多媒体和游戏开发人员,大专院校相关专业师生和程序设计爱好者。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无防伪标签者不得销售。

书 名: Direct3D 原理与 API 参考(上)

作 者: 武永康

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编:100084)

<http://www.tup.tsinghua.edu.cn>

印 刷 者: 世界知识印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 23 字数: 543 千字

版 次: 2001 年 10 月第 1 版 2001 年 10 月第 1 次印刷

书 号: ISBN 7-302-04774-X/TP·2828

印 数: 0001~5000

定 价: 32.00 元

前 言

Direct3D 是 DirectX 的一个重要组件,它能使游戏和交互式三维图形在使用 Microsoft Windows 的计算机上运行。Direct3D 用与设备无关的方法对 3D 视频加速硬件进行设备相关的访问。可以用两种模式使用 Direct3D: 立即模式(Immediate Mode)和保留模式(Retained Mode)。Direct3D 立即模式是一种低级的三维应用程序编程接口,对于需要把游戏和其他高性能的多媒体应用程序移植到 Microsoft Windows 操作系统的开发者来说是一种理想的工具。对于在底层与加速硬件通信的应用程序来说,立即模式使用的是一种与设备无关的方法。

Direct3D 保留模式构建于立即模式的顶端之上,封装了立即模式接口。保留模式是一种高级的三维应用程序编程接口,适用于要求快速开发或者要求保留模式对层级和动画的内建支持帮助的程序员。

Direct3D 有下列高级性能:

- 可切换的深度缓冲技术(使用 z 缓存或者 w 缓存)。
- Flat 和 Gouraud 明暗处理。
- 多种光源和类型。
- 完整的材质和纹理支持,包括多纹理映像。
- 强大的软件仿真驱动程序。
- 变换和剪裁。
- 硬件无关性。
- 完全支持 Windows 95、Windows 98 和 Windows 2000。
- 支持 Intel MMX 系统结构。

Direct3D 立即模式对“世界”的管理基于顶点、多边形及控制它们的命令。它允许直接访问变换、光照和光栅化三维图形流水线。如果硬件不提供加速渲染功能,那么 Direct3D 就给应用程序提供一种强大的软件仿真功能。凡是现有三维应用程序的开发者 and 需要通过在应用程序和硬件之间使档次尽可能达到最高性能的开发者,都应使用立即模式而不是保留模式。

Direct3D 立即模式提供了一些简单的方法来设置并渲染一个三维场景。关键的渲染方法集就是 DrawPrimitive 方法。它使应用程序能够只用一个方法调用就可以渲染一个场景中的一个或多个对象。

Direct3D 立即模式对三维加速硬件提供了一种低开销的连接。这种低开销连接的代价就是对变换和光照必须提供显式调用,必须提供所有必要的矩阵,还必须确定所提供硬

件的类型及其性能。

鉴于以上理由,本书以 DirectX 7.0 的 Direct3D 立即模式应用程序编程接口为主来介绍 Direct3D。Direct3D 立即模式支持在 C/C++ 和 Visual Basic 两种语言环境下开发 Direct3D 应用程序,因此对于这两种环境下编程的异同,也分别做了详细讨论。

目 录

第 1 章 计算机三维图形的基本概念	1
1.1 三维坐标系和几何体	1
1.1.1 三维坐标系	1
1.1.2 三维图元	2
1.1.3 三角形光栅化规则	3
1.2 明暗处理	4
1.2.1 明暗处理方式	5
1.2.2 比较明暗处理方式	5
1.2.3 设置明暗处理方式	6
1.2.4 平面和顶点的法线向量	7
1.2.5 三角形内插	11
1.3 矩阵和变换	12
1.3.1 矩阵	12
1.3.2 三维变换	12
第 2 章 Direct3D 立即模式基础	17
2.1 Direct3D 立即模式的层次结构	17
2.1.1 Direct3D 立即模式对象类型	18
2.1.2 Direct3D 立即模式的 COM 接口	19
2.2 DirectX 7.0 的 Direct3D 立即模式的改进	20
2.3 Direct3D 和 DirectDraw	21
2.3.1 DirectDraw 对象 和 Direct3D	21
2.3.2 Direct3D 接口	22
2.3.3 访问 Direct3D 接口	23
2.3.4 创建附属于 Direct3D 的对象	25
2.4 Direct3D 设备	25
2.4.1 Direct3D 设备的概念	25
2.4.2 Direct3D 设备的类型	26
2.4.3 设备接口	29
2.4.4 设备状态	30
2.4.5 使用设备	56
2.4.6 AGP 表面和 Direct3D 设备	80

第 3 章 几何流水线	81
3.1 流水线概述	81
3.2 世界变换	82
3.2.1 什么是世界变换的概念	82
3.2.2 设置一个世界矩阵	83
3.2.3 性能优化	85
3.3 观察变换(View Transformation)	85
3.3.1 观察变换的概念	85
3.3.2 设置一个观察矩阵	87
3.3.3 性能优化	88
3.4 投影变换	89
3.4.1 观察截锥	89
3.4.2 投影变换的概念	90
3.4.3 设置一个投影矩阵	92
3.4.4 一个“w-友好的”投影矩阵	93
3.5 视口和裁剪	94
3.5.1 什么是视口	94
3.5.2 视口矩形	94
3.5.3 裁剪体	96
3.5.4 视口缩放	96
3.5.5 使用视口	97
3.6 光栅化处理器	101
第 4 章 光照和材质	102
4.1 有关光照和材质的基本概念	102
4.1.1 Direct3D 光照模型与自然光	103
4.1.2 光和材质的颜色值	103
4.1.3 直射光与环境光	104
4.1.4 启用和禁止光照引擎	105
4.2 光照	105
4.2.1 介绍光照对象	105
4.2.2 光照属性	107
4.2.3 使用光照	110
4.3 材质	115
4.3.1 材质的概念	115
4.3.2 材质属性	116
4.3.3 使用材质	118
4.4 Direct3D 光照的数学原理	121

4.4.1 光的颜色类型和光源	121
4.4.2 光随距离的衰减	122
4.4.3 反射比模型	123
4.4.4 聚光灯衰落模型	127
4.4.5 烟雾效果	129
第 5 章 纹理	130
5.1 纹理的基本概念	130
5.1.1 纹理的概念	130
5.1.2 纹理的寻址方式	131
5.1.3 纹理接口和纹理句柄	136
5.1.4 调色板化的纹理	137
5.2 纹理坐标	138
5.2.1 纹理坐标的概念	138
5.2.2 直接把纹理元素映射为像素	141
5.2.3 纹理坐标格式	142
5.2.4 纹理坐标处理	144
5.3 纹理表面对象	152
5.3.1 获得一个纹理表面对象	153
5.3.2 用纹理表面渲染	153
5.4 纹理过滤	154
5.4.1 最近点采样	155
5.4.2 线性纹理过滤	157
5.4.3 非均匀性纹理过滤	158
5.4.4 使用多纹理映像的纹理过滤	159
5.5 纹理包装	164
5.5.1 纹理包装的概念	165
5.5.2 使用纹理包装	167
5.6 纹 理 混 合	167
5.6.1 Alpha 纹理混合	168
5.6.2 多次纹理混合	169
5.6.3 多纹理混合	169
5.6.4 使用纹理的光映射图	174
5.7 纹理压缩	179
5.8 自动纹理管理	180
5.9 纹理的硬件考虑	181
第 6 章 顶点格式	183
6.1 关于顶点格式	183

6.2	未变换、无光照的顶点	185
6.3	未变换、有光照的顶点	187
6.4	已变换、有光照的顶点	189
6.5	跨距顶点格式	191
第 7 章	深度缓冲技术	194
7.1	深度缓冲器的概念	194
7.2	使用深度缓冲器	196
7.2.1	查询对深度缓冲器的支持	196
7.2.2	创建一个深度缓冲器	197
7.2.3	启用深度缓冲器	199
7.2.4	清除深度缓冲器	199
7.2.5	改变深度缓冲器的写访问	200
7.2.6	改变深度缓冲器的比较功能	201
7.2.7	使用 α - 偏离	201
7.3	模板缓冲器	202
7.3.1	模板缓冲器的概念	202
7.3.2	模板缓冲器如何工作	202
7.3.3	定制模板缓冲器	203
7.4	顶点缓冲器	203
7.4.1	顶点缓冲器的概念	204
7.4.2	顶点缓冲器描述	204
7.4.3	设备类型和顶点处理需求	206
7.4.4	使用顶点缓冲器	207
第 8 章	通用技术和特殊效果	220
8.1	烟雾	220
8.1.1	烟雾效果简介	220
8.1.2	烟雾计算公式	221
8.1.3	烟雾的参数	223
8.1.4	烟雾混合	224
8.1.5	烟雾的颜色	224
8.1.6	像素烟雾	225
8.1.7	顶点烟雾	229
8.2	几何混合	233
8.2.1	关于几何混合	233
8.2.2	混合变换和渲染状态	235
8.2.3	混合加权	236
8.2.4	使用几何混合	237

8.3 用户定义的裁剪位面	240
8.3.1 用户定义的裁剪位面	240
8.3.2 使用用户定义的裁剪位面	242
8.4 环境映射	244
8.4.1 环境映射的概念	245
8.4.2 立方体环境映射	245
8.4.3 球形环境贴图	257
8.5 凹凸映像	258
8.5.1 凹凸映像	259
8.5.2 凹凸映像贴图的像素格式	260
8.5.3 凹凸映像公式	261
8.5.4 使用凹凸映像	261
8.6 布告板	267
8.7 云、烟和雾化尾迹	267
8.8 火焰、闪烁和爆炸	268
8.9 运动模糊	269
8.10 彩色光	270
8.11 防锯齿技术	270
8.11.1 边沿防锯齿	270
8.11.2 全屏幕防锯齿	271
第 9 章 Direct3D 立即模式实例 (C/C++)	273
9.1 渲染一个三角形	273
9.1.1 创建一个窗口	273
9.1.2 初始化系统对象	274
9.1.3 初始化场景	279
9.1.4 监控系统消息	281
9.1.5 渲染并显示场景	282
9.1.6 关闭	284
9.1.7 处理窗口移动	285
9.1.8 调整窗口尺寸的处理	286
9.2 增加一个深度缓冲器	287
9.2.1 枚举深度缓冲器格式	287
9.2.2 创建深度缓冲器	288
9.2.3 附加深度缓冲器	289
9.2.4 启用深度缓冲器	290
9.3 使用交替的图元样式	290
9.3.1 初始化场景几何体	290

9.3.2 渲染和显示场景	292
9.4 使用设备枚举	296
9.4.1 初始化和完成设备枚举	297
9.4.2 选择一种枚举设备	300
9.5 使用纹理贴图	301
9.5.1 准备一个纹理表面	302
9.5.2 创建一个纹理表面	305
9.5.3 渲染一个有纹理图元	307
第 10 章 Direct3D 立即模式实例(Visual Basic)	308
10.1 渲染一个三角形	308
10.1.1 创建一个窗体	309
10.1.2 初始化系统对象	309
10.1.3 初始化场景	313
10.1.4 执行渲染循环	316
10.1.5 关闭	318
10.2 增加一个深度缓冲器	319
10.2.1 枚举深度缓冲器格式	319
10.2.2 创建深度缓冲器	320
10.2.3 附加深度缓冲器	321
10.2.4 启用深度缓冲器	321
10.3 使用交替的图元样式	322
10.3.1 初始化场景几何体	322
10.3.2 渲染和显示场景	324
10.4 使用设备枚举	327
10.4.1 初始化和完成设备枚举	327
10.4.2 选择一种枚举设备	329
10.5 使用纹理贴图	331
10.5.1 创建一个纹理表面	332
10.5.2 渲染一个有纹理图元	334
第 11 章 Direct3DX 实用库	335
11.1 关于 Direct3DX 实用库	335
11.2 使用 Direct3DX 实用库	336
11.3 Direct3DX 实用库的层次结构	336
11.3.1 Direct3DX 实用库层次结构概述	336
11.3.2 Direct3DX 实用库的接口和对象类型	337
11.4 Direct3DX 实用库基本原理	338
11.4.1 Direct3DX 和 Direct3D	339

11.4.2	Direct3DX 设备环境	340
11.4.3	Direct3DX 设备	341
11.4.4	Direct3DX 矩阵堆栈	344
11.4.5	硬件加速级别	346
11.4.6	Sprite 支持	347
11.4.7	纹理处理支持	350
11.4.8	简单的几何体形状	354
11.4.9	3D 数学支持	355

第 1 章 计算机三维图形的基本概念

本章描述在合成三维图形的程序之前必须了解的一些概念。它并讨论体系结构的细节,也不对特定的 Direct3D 部件做深入分析。如果读者在产生三维图形方面已有经验,那么只需浏览本章有关 Direct3D 的内容即可。

本章主要内容:

- 三维坐标系和几何体
- 明暗处理
- 矩阵和变换

1.1 三维坐标系和几何体

编制应用程序要通晓三维几何原理。本节介绍创建三维场景所需了解的主要几何概念:

- 三维坐标系
- 三维图元
- 三角形光栅化规则

1.1.1 三维坐标系

典型的三维图形应用程序使用两种笛卡尔坐标系:左手和右手系统,如图 1.1 和图 1.2 所示。

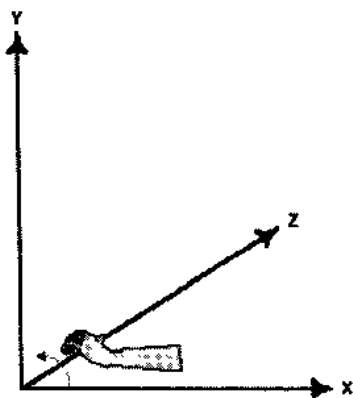


图 1.1 左手系统

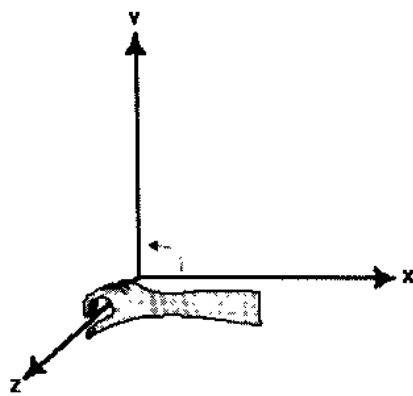


图 1.2 右手系统

Direct3D 使用左手坐标系。如果移植一个基于右手坐标系的应用程序,只需把要传送给 Direct3D 的数据做两项简单的改变即可:

翻转三角顶点次序,使系统依顺时针方向从前到后遍历它们。换句话说,如果顶点是 v_0, v_1, v_2 ,那么就按 v_0, v_2, v_1 的次序传送给 Direct3D。

使用观察矩阵,通过在 z 方向减 1 来缩放世界空间。为此,反转为观察矩阵而使用的 D3DMATRIX 结构中的 -31、-32、-33 和 -34 成员的符号(同样,Visual Basic 应用程序可以在 D3DMATRIX 类型的相应成员中执行这种操作)。

在一种三维坐标系中所定义对象上执行的基本操作是平移(translation)、旋转(rotation)和缩放(scaling)。可以通过组合这些基本变换来创建一个变换矩阵。

1.1.2 三维图元

三维图元就是形成单个三维实体的顶点集合。最简单的图元就是在一个三维坐标系中点列表(point list)的点集合。三维 D 图元通常是多边形,该多边形是一个封闭的、至少由 3 个顶点描绘的三维图形。最简单的多边形就是三角形。Direct3D 用三角形组成大多数多边形,因为一个三角形中的 3 个顶点肯定是共面的。渲染非平面顶点是低效的。可以通过组合多个三角形来形成大而复杂的多边形和网格。

图 1.3 为一个立方体,它的每个面都由两个三角形构成。整个三角形组合起来就形成了一个立方体图元。可以把纹理和材质贴到这个图元的表面,使它们看起来像是纯固体形状。

也可以使用三角形创建表面呈现平滑曲线的图元。图 1.4 为如何用三角形模拟一个球体。在使用了材质并渲染它以后,该球体看起来像是弯曲的。如果使用 Gouraud 明暗处理法,则更显真实。

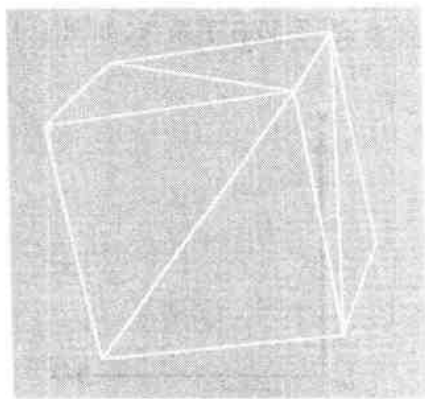


图 1.3 立方体的构成

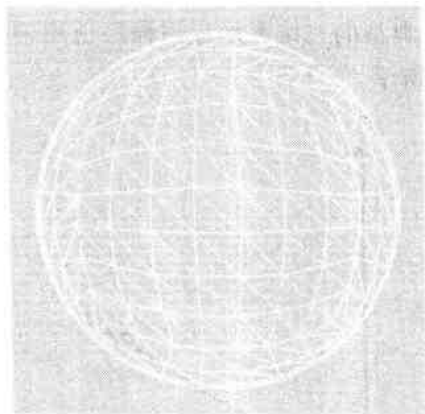


图 1.4 球体的构成

1.1.3 三角形光栅化规则

对顶点所指定的点常常不能精确地与屏幕上的像素相匹配,这时 Direct3D 应用三角形光栅化规则,来确定把哪些像素用于给定的三角形。Direct3D 填充几何体时使用左上角(top-left)填充约定,它与在 GDI 和 OpenGL 中对矩形所用的约定相同。在 Direct3D 中,像素的中心是决定性的点。如果像素的中心在三角形内部,则该像素就是这个三角形的一部分。像素中心用整数坐标。

这个对 Direct3D 所使用的三角形光栅化规则的描述没有必要用于所有可用硬件。通过测试可以发现实现这些规则时的微小差别。

图 1.5 所示为一个左上角坐标为(0,0)、右下角坐标为(5,5)的矩形。

使用左上角(top-left)填充约定, top 表示水平跨距的垂直位置,而 left 表示在跨距里像素的水平位置。一条边不能是顶边,除非它是水平的——一般情况下,大多数三角形只有左边和右边,如图 1.6 所示。

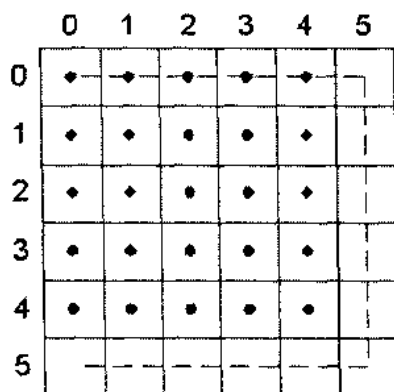


图 1.5 屏幕上的矩形

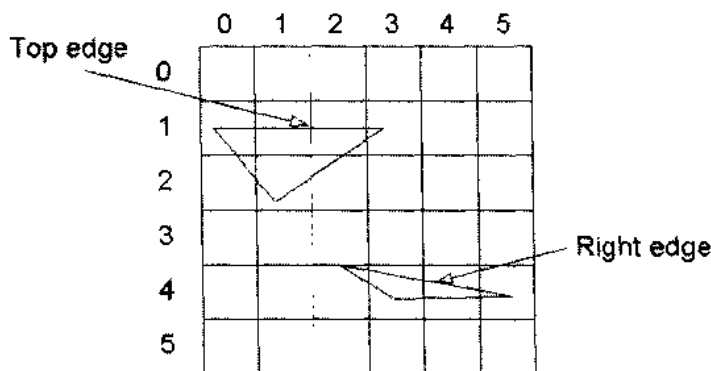


图 1.6 屏幕上的三角形

左上角(top-left)填充约定确定一个三角形通过一个像素的中心时, Direct3D 所应采用的动作。图 1.7 给出两个三角形,一个三角形的顶点坐标分别为(0, 0)、(5, 0)和(5, 5),另一个三角形的顶点坐标分别为(0, 5)、(0, 0)和(5, 5)。在这种情况下,第 1 个三角形得到 15 个像素,而第 2 个三角形只得到 10 个像素,因为共享边是第 1 个三角形的左边。

如果定义了一个左上角坐标为(0.5, 0.5)、右下角坐标为(2.5, 4.5)的矩形,则这个矩形的中心点坐标为(1.5, 2.5)。当用 Direct3D 光栅化处理器铺嵌这个矩形时,每个像素的中心都在 4 个三角形内,因此不必使用左上角(top-left)填充约定。如图 1.8 所示,按照

Direct3D 在该矩形中三角形所包含的像素标记它们。

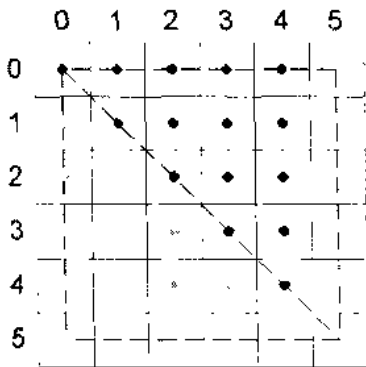


图 1.7 屏幕上两个三角形

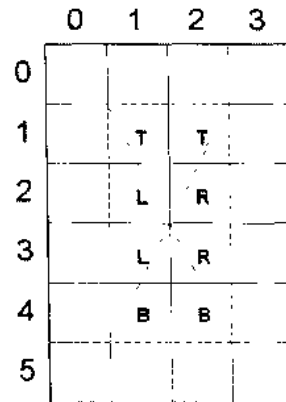


图 1.8 用矩形中三角形所包含的像素标记

如果移动上述的矩形,使其左上角坐标为(1.0,1.0),右下角坐标为(3.0,5.0)且中心点坐标为(2.0,3.0), Direct3D 使用左上角(top-left)填充约定。在这个矩形中的大多数像素都位于在两个或多个三角形之间的边沿,如图 1.9 所示。

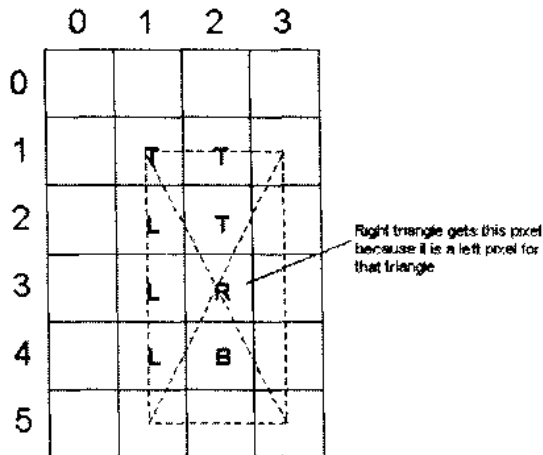


图 1.9 像素都跨骑在两个三角形之间边沿

1.2 明暗处理

本节描述在 Direct3D 中控制三维多边形明暗的处理技术:

- 明暗处理方式。
- 比较明暗处理方式。

- 设置明暗处理方式。
- 表面和顶点的法线向量。
- 三角形内插。

1.2.1 明暗处理方式

对于多边形的明暗处理影响其外观。明暗处理确定多边形面上任意点处的颜色和光照的亮度。Direct3D 当前支持两种明暗处理方式：

- Flat 明暗处理方式；
- Gouraud 明暗处理方式。

1. Flat 明暗处理方式

用 Flat 明暗处理方式渲染一个多边形时,把这个多边形第 1 个顶点上的颜色用作整个多边形的颜色。用 Flat 明暗处理方式所渲染的三维对象,如果多边形不共面时,就会在它们之间有明显的边沿。

图 1.10 所示为用 Flat 明暗处理方式渲染的茶壶。每个多边形的轮廓都清晰可见。Flat 是一种在计算上代价最小的明暗处理方式。



图 1.10 Flat 明暗处理效果



图 1.11 Gouraud 明暗处理的效果

2. Gouraud 明暗处理方式

当 Direct3D 用 Gouraud 明暗处理方式渲染多边形时,它使用顶点法线和光照参数来计算每个顶点的颜色,然后在多边形面之间插补颜色。这种插补是线性的。例如,如果顶点 1 的颜色中红成分的值为 0.8,而顶点 2 的颜色中红成分的值 0.4,利用 Gouraud 明暗处理方式和 RGB 颜色模型,Direct3D 光照模块将把红成分为 0.6 的值分配给这两个顶点间线段中点的那个像素。

图 1.11 所示为 Gouraud 明暗处理的结果。该茶壶由许多平面和三角形组成,但经 Gouraud 明暗处理后使对象的表面显得弯曲和平滑。

Gouraud 明暗处理也可以用于渲染有明显棱角的对象。

1.2.2 比较明暗处理方式

使用 Flat 明暗处理方式,棱锥相邻面之间就会出现明显的棱,如图 1.12 所示,而使用 Gouraud 明暗处理方式,将在这些棱之间插补明暗处理值,于是最后的外观就会是弯曲平滑的。