

湖南科技大学课程教案

(章节、专题首页)

授课教师: 王志喜

职称: 副教授

单位: 计算机科学与工程学院

课 程 名 称	计算机图形图像技术
章节、专题	三维场景的真实感绘制
教学目标及 基 本 要 求	(1) 掌握最常用的可见面判别算法。 (2) 掌握基本的光照模型和典型的面绘制算法。
教 学 重 点	深度缓冲器算法, 基本光照模型, 多边形绘制算法
教 学 难 点	深度缓冲器算法, 基本光照模型, 多边形绘制算法
教学内容与 时 间 分 配	(1) 深度缓冲器算法 (0.5课时) (2) 光源与基本光照模型 (1课时) (3) 多边形面绘制算法 (0.5课时) 共计2课时。
习 题	第7.7.1节 (基础知识题)。

第7章 三维场景的真实感绘制

7.1 概述

三维场景的真实感绘制包括下列2方面内容。

- 可见面判别算法。用于判断场景中各面片的可见性。
- 光照模型与面绘制算法。用于计算场景中物体表面所有位置的光强度。

7.1.1 可见面判别算法

1. 分类

可见面判别算法也可称为隐藏面消除算法，可以分为2类。

- 物空间算法。将场景中各物体和各组成部分相互进行比较，以判别哪些面可见。例如后向面判别算法。
- 像空间算法。在投影平面上逐点判断各像素对应的可见面。例如深度缓冲器算法。

2. 常用算法

常用的可见面判别算法有

- 后向面判别算法（物空间算法）
- 深度缓冲器算法（也叫z-buffer算法，像空间算法，OpenGL和Matlab使用了该算法）
- 扫描线算法（像空间算法）
- 深度排序算法（也叫画家算法，像空间算法，Matlab使用了该算法）

本章只介绍图形软件包中最常用深度缓冲器算法。

7.1.2 光照模型与面绘制算法

- 光照模型（也称明暗模型）。用于物体表面某点处的光强度计算。
- 面绘制算法（也称渲染算法）。通过光照模型中的光强度计算，确定场景中物体表面所有投影像素点的光强度。

7.2 深度缓冲器算法

7.2.1 基本思想

将投影平面上每个像素所对应面片的深度进行比较，然后取最近面片的属性值作为该像素的属性值。

7.2.2 深度比较的实现

在 PC 中，多边形面片上的每个点 (x, y, z) 均对应于观察面上的像素 (x, y) 。因此，对观察面上的每个像素 (x, y) ，物体的深度比较可以通过 z 值的比较实现。如图 7-1 所示。

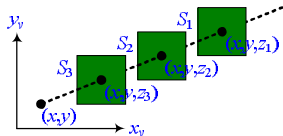


图 7-1 深度比较的实现

7.2.3 在NC中实现深度缓冲器算法

1. z 值范围

在OpenGL等图形软件包中 z 值的范围是 -1 （对应近裁剪面）到 1 （对应远裁剪面）之间。

2. 缓冲器

共2个，分别是深度缓冲器和刷新缓冲器。

- 深度缓冲器。保存面片上各像素所对应的深度值，初始化为 1 （最大深度）。
- 刷新缓冲器。保存各点的属性值，初始化为背景属性。

3. 算法步骤

(1) 初始化深度缓冲器和刷新缓冲器的所有单元。

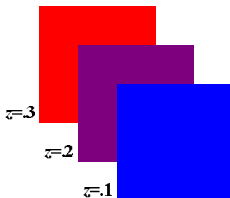
$$\text{depth}(x, y) = 1, \quad \text{frame}(x, y) = I_{\text{back}} \quad (\text{背景属性})$$

(2) 将每个多边形面片上各点的深度与深度缓冲器中记录的相应深度值比较，以确定这些点的可见性。每次处理一个面片。

- 计算多边形面片上各点的深度 z 。
- 若 $z < \text{depth}(x, y)$ ，则 $\text{depth}(x, y) = z$ ， $\text{frame}(x, y) = I_{\text{surf}}(x, y)$ (面片属性)。

(3) 处理完毕以后，深度缓冲器中保存的是可见面的深度值，刷新缓冲器中保存的是相应的属性值。

图7-2演示了上述步骤。



[1] 三个矩形面片

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

[2] 初始化

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	2	2	2	2	2	2	1	1
1	1	2	2	2	2	2	2	1	1
1	1	2	2	2	2	2	2	1	1
1	1	2	2	2	2	2	2	1	1
1	1	2	2	2	2	2	2	1	1
1	1	2	2	2	2	2	2	1	1
1	1	2	2	2	2	2	2	1	1
1	1	1	1	1	1	1	1	1	1

[3] 处理紫色面片

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	2	2	2	2	2	2	1	1
1	1	2	2	2	2	2	2	1	1
1	1	2	2	1	1	1	1	1	1
1	1	2	2	1	1	1	1	1	1
1	1	2	2	1	1	1	1	1	1
1	1	2	2	1	1	1	1	1	1
1	1	2	2	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

[4] 处理蓝色面片

3	3	3	3	3	3	1	1	1	1
3	3	3	3	3	3	1	1	1	1
3	3	2	2	2	2	2	2	1	1
3	3	2	2	2	2	2	2	1	1
3	3	2	2	1	1	1	1	1	1
3	3	2	2	1	1	1	1	1	1
1	1	2	2	1	1	1	1	1	1
1	1	2	2	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

[5] 处理红色面片

图 7-2 深度缓冲器算法步骤演示

7.2.4 深度值的计算

1. 每条扫描线上的深度值计算

如图7-3所示。 (x, y) 处的深度值为

$$z = \frac{-Ax - By - D}{C}$$

$(x+1, y)$ 处的深度值为

$$\begin{aligned} z' &= \frac{-A(x+1) - By - D}{C} \\ &= z - A/C \end{aligned}$$

由上可知，对于每条扫描线，后一点的深度值可以通过前一点的深度值加上一个常数得到。

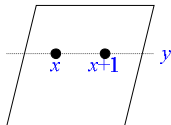


图 7-3 扫描线上的深度值

2. 扫描线与多边形左边的交点和深度

对于多边形的每条左边，由最上方顶点出发，沿该边递推计算交点的 x 坐标和深度值。如图7-4所示。设 m 为该边的斜率。易知， $x' = x - 1/m$ 。 (x, y) 处的深度值为

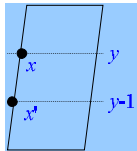
$$z = \frac{-Ax - By - D}{C}$$

$(x', y-1)$ 处的深度值

$$\begin{aligned} z' &= \frac{-A(x - 1/m) - B(y - 1) - D}{C} \\ &= z + \frac{A/m + B}{C} \end{aligned}$$

从而得到扫描线 $y-1$ 与多边形左边的交点和深度为

$$\begin{cases} x' = x - 1/m \\ z' = z + \frac{A/m + B}{C} \end{cases}$$



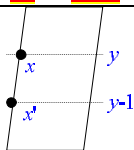
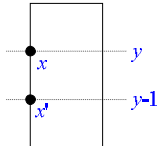


图 7-4 扫描线与边的交点

若左边为垂直边（如图7-5所示），则

$$\begin{cases} x' = x \\ z' = z + B/C \end{cases}$$


The diagram shows a vertical rectangle. Two horizontal dashed lines extend from the left side of the rectangle. The top line is labeled 'y' on the right and has a point on the rectangle labeled 'x' in blue. The bottom line is labeled 'y-1' on the right and has a point on the rectangle labeled 'x'' in blue.

图 7-5 垂直边的情况

7.3 光源

7.3.1 光源的含义

1. 含义

所有发射辐射能量的物体都可以称为光源。

2. 举例

- 发光体：太阳、灯泡。
- 反射光源：墙壁、月亮。

7.3.2 光源的分类

主要有三类：点光源、分布式光源和环境光。

1. 点光源

- (1) 特征。光线向四周发散。
- (2) 举例。下列光源可抽象为点光源。
 - 光源远远小于物体，如小灯泡。
 - 光源远离物体，如太阳。

2. 分布式光源

- (1) 特征。需要计算光源外表面各点共同产生的光照。
- (2) 举例。下列光源可抽象为分布式光源。
 - 跟物体相比，光源不够小，如日光灯。
 - 光源靠近物体，如台灯。

3. 环境光

- (1) 含义。从不同物体表面产生的反射光的统一照明。
- (2) 模拟方法。改变一个场景的基准光亮度。
- (3) 大小。用 I_a 表示。
- (4) 特征。环境光主要有下列特征。
 - 每个物体表面得到同样大小的光照。
 - 反射光与观察方向和物体的朝向无关。
 - 反射光强度决定于各个表面的材料属性。

7.3.3 关于反射

1. 反射的分类

- 漫反射。粗糙物体的表面往往将反射光向各个方向散射。
- 镜面反射。当观察一个光照下的光滑物体表面时，可能在某个观察方向看到高光或强光。

2. 入射角和反射角

- 入射角。入射光线与物体表面法向量之间的夹角。
- 反射角。反射光线与物体表面法向量之间的夹角。

如图7-6所示。

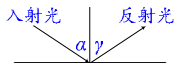


图 7-6 入射角和反射角

7.4 基本光照模型

只考虑环境光和点光源。

7.4.1 漫反射

1. 漫反射系数

入射光中被反射部分的百分比，用 k_d 表示。

2. 环境光的漫反射

若环境光强度为 I_a ，则反射光强度为 $I_{\text{ambdiff}} = k_d I_a$ 。

3. 点光源的漫反射

相关的几何量如图7-7所示。

- θ 是入射角。
- \vec{N} 是物体表面的单位法向量。
- \vec{L} 是物体表面到点光源的单位向量。

若点光源强度为 I_l ，则反射光强度为

$$I_{l,\text{diff}} = k_d I_l \cos \theta = k_d I_l (\vec{N} \cdot \vec{L})$$

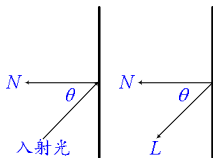


图 7-7 入射线向量和表面法向量

【注】本章出现的正弦、余弦和内积都是非负的，否则，直接用0代替。

4. 完整的漫反射方程

$$I_{\text{diff}} = k_a I_a + k_d I_l (\vec{N} \cdot \vec{L})$$

其中， k_a 是环境光漫反射系数， k_d 是点光源漫反射系数。

5. 不同反射系数对应的效果演示

图7-8演示了不同反射系数对应的效果。其中，横向反映点光源漫反射系数的变化，纵向反映环境光漫反射系数的变化（0、0.25、0.5、0.75、1）。

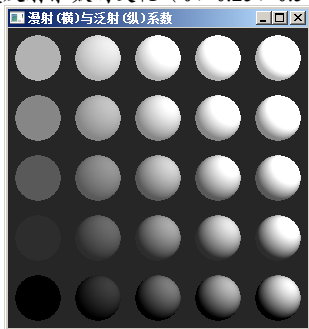


图 7-8 不同反射系数对应的效果演示

7.4.2 镜面反射和Phong光照模型

1. 镜面反射

(1) 含义。当观察一个光照下的光滑物体表面时，可能在某个观察方向看到高光或强光。如图7-9所示。

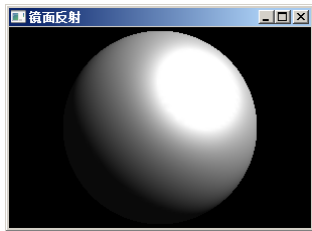


图 7-9 镜面反射

(2) 特性。镜面反射主要有下列特征（如图7-10所示）。

- 入射光线、表面法向量、反射光线共面。
- 入射光线和反射光线位于表面法向量的两侧。
- 反射角=入射角。

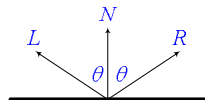


图 7-10 镜面反射的特性

2. Phong模型的相关几何量

如图7-11所示。

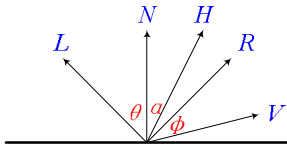


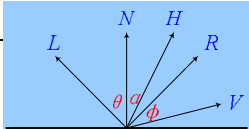
图7-11 Phong模型

- \vec{L} 是指向点光源的单位向量。
- \vec{N} 是物体表面法向量。
- \vec{V} 是指向视点的单位向量。
- \vec{R} 是反射方向的单位向量。
- ϕ 是 \vec{R} 与 \vec{V} 的夹角。

3. Phong模型的计算公式

$$I_{\text{spec}} = k_s I_l \cos^{n_s} \alpha = k_s I_l (\bar{N} \cdot \bar{H})^{n_s}$$

其中， $\bar{H} = (\bar{L} + \bar{V}) / |\bar{L} + \bar{V}|$ 是 \bar{L} 与 \bar{V} 的半角向量， α 是 \bar{N} 和 \bar{H} 的夹角。



4. Phong模型的相关物理量

- I_l 是光源强度， I_{spec} 是反射光强度。
- k_s 是镜面反射系数。取值在 0~1 之间，与物体表面材质相关，仅适用于不透明材质。
- n_s 是光泽度或镜面反射指数。反映镜面反射光的聚集程度，由物体表面材料属性决定。 n_s 越大，反射光越集中在反射方向附近。光滑表面的 n_s 比较大（如100或更大），粗糙表面的 n_s 比较小（如1或更小），理想反射器的 n_s 为无限大。

\bar{L} 是指向点光源的单位向量。
 \bar{N} 是物体表面法向量。
 \bar{V} 是指向视点的单位向量。
 \bar{R} 是反射方向的单位向量。
 ϕ 是 \bar{R} 与 \bar{V} 的夹角。

7.4.3 漫反射和镜面反射的合并

考虑环境光、漫反射和镜面反射。

1. 单个点光源

$$I = k_a I_a + k_d I_l (\vec{N} \cdot \vec{L}) + k_s I_l (\vec{N} \cdot \vec{H})^{n_s}$$

2. 多个光源

$$I = k_a I_a + \sum_{i=1}^n I_{l_i} \left[k_d (\vec{N} \cdot \vec{L}_i) + k_s (\vec{N} \cdot \vec{H}_i)^{n_s} \right]$$

7.4.4 强度衰减

1. 简单模型

(1) 方法。光线强度按照因子 $1/d^2$ 衰减，即 $f(d) = 1/d^2$ ，其中， d 是光线经过的路程长度。

(2) 缺陷。过于简单，不能总产生真实感图形，因为可能有 $f(d) > 1$ 。

2. 改进模型

(1) 公式。改进后的计算公式为 $f(d) = 1/(a_0 + a_1d + a_2d^2)$ 。

(2) 调整参数得到不同的光照效果。调整 a_0, a_1, a_2 可以得到不同的光照效果。

- a_0 可以防止当 d 太小时 $f(d)$ 太大。
- 调节 a_0, a_1, a_2 和物体表面参数，可以防止反射光的强度超过允许上限。

(3) 限制衰减函数的范围。使用公式 $f(d) = \min(1, 1/(a_0 + a_1d + a_2d^2))$ 计算。

3. 改进后的基本光照模型

$$I = k_a I_a + \sum_{i=1}^n f(d_i) I_{l_i} \left[k_d (\vec{N} \cdot \vec{L}_i) + k_s (\vec{N} \cdot \vec{H}_i)^{n_s} \right]$$

7.4.5 颜色

1. 描述颜色的方法

每个颜色用 R 、 G 、 B 三个分量表示。用 R 、 G 、 B 三个分量表示光源强度和物体表面颜色，根据光照模型计算反射光线中的 R 、 G 、 B 分量。

2. 设置表面颜色的方法

有多种设置表面颜色的方法，这里只介绍将反射系数表示为三元向量的方法。OpenGL使用了这种方法。

(1) 三元向量的表示。表示为 (k_{dR}, k_{dG}, k_{dB}) ，其中， $k_{dR}, k_{dG}, k_{dB} \in (0,1]$ 。

(2) 计算方法。使用下列公式计算物体表面颜色。

$$I_R = k_{aR}I_{aR} + \sum_{i=1}^n f(d_i)I_{lR_i} \left[k_{dR}(\overline{N} \cdot \overline{L}_i) + k_{sR}(\overline{N} \cdot \overline{H}_i)^{n_s} \right]$$

$$I_G = k_{aG}I_{aG} + \sum_{i=1}^n f(d_i)I_{lG_i} \left[k_{dG}(\overline{N} \cdot \overline{L}_i) + k_{sG}(\overline{N} \cdot \overline{H}_i)^{n_s} \right]$$

$$I_B = k_{aB}I_{aB} + \sum_{i=1}^n f(d_i)I_{lB_i} \left[k_{dB}(\overline{N} \cdot \overline{L}_i) + k_{sB}(\overline{N} \cdot \overline{H}_i)^{n_s} \right]$$

3. OpenGL中的相关函数调用

主要有下列调用形式。

- `glMaterialfv(face, GL_AMBIENT, mater)`。指定face面的环境光漫反射系数（RGBA，数组）。
- `glMaterialfv(face, GL_DIFFUSE, mater)`。指定face面的点光源漫反射系数（RGBA，数组）。
- `glMaterialfv(face, GL_SPECULAR, mater)`。指定face面的镜面反射系数（RGBA，数组）。
- `glMaterialfv(face, GL_SHININESS, mater)`。指定face面的镜面反射指数（单值，数组）。
- `glMaterialf(face, GL_SHININESS, mater)`。指定face面的镜面反射指数（单值）。

7.4.6 透明效果[※]

1. 对问题的简化

不考虑折射导致的光线路径平移（如图7-12[1]所示）。

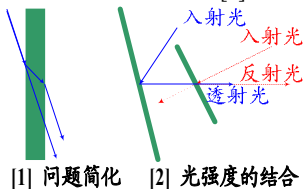


图7-12 透明物体表面光强度

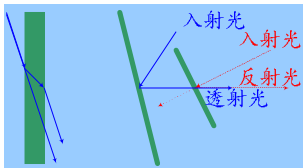
2. 物体表面光强度的计算

如图7-12[2]所示，透明物体表面的光强度是透射光强度和反射光强度的结合强度。

(1) 计算公式。 $I = (1 - k_t)I_{\text{refl}} + k_t I_{\text{trans}}$ 。

(2) 有关参数说明。

- I_{refl} 是反射光强度。对透明物体用反射光照模型计算。
- I_{trans} 是透射光强度。对后面的物体用反射光照模型计算。
- k_t 是透明系数。介于0与1之间，表示有多少背景光被透射。高度透明物体，接近于1。不透明物体，接近于0。



3. 透明效果的实现

通常使用可见面判别算法来实现。例如，使用深度缓冲器算法的实现方法为如下步骤。

- 处理不透明物体，确定可见不透明表面的深度。
- 将透明物体的深度值与深度缓冲器中的值比较。
- 若透明物体的表面是可见的，则计算反射光强度并且和刷新缓冲器中的光强度结合。

7.5 多边形面绘制算法

多边形面绘制算法用于利用扫描线算法从基本光照模型来实现多边形面片的绘制。基本方法有两种。

- 平面明暗处理。将每个多边形面片用单一光强度绘制。
- 光滑明暗处理。利用扫描线算法得到面片上各点的光强度，如Gouraud明暗处理、Phong明暗处理等。

在OpenGL中，使用`glShadeModel(GL_FLAT)`选择平面明暗处理，使用`glShadeModel(GL_SMOOTH)`选择光滑明暗处理（Gouraud明暗处理）。两种方法的比较如图7-13所示。

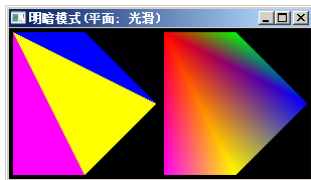


图7-13 Gouraud明暗处理与平面明暗处理

7.5.1 平面明暗处理

1. 方法

每个多边形面片均使用一个光强度，面片上所有点均使用相同的光强度显示。如图7-13所示。

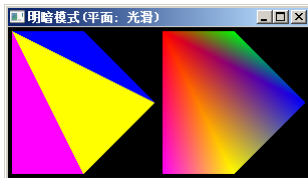
2. 可以准确绘制的物体

用平面明暗处理可以准确绘制的物体应满足下列条件。

- 物体本身是一个多面体。
- 所有照明光源离物体足够远，使得 $\vec{N} \cdot \vec{L}$ 和衰减函数对物体表面来讲是一个常数。
- 视点离物体足够远，使得 $\vec{V} \cdot \vec{R}$ 对物体表面来讲是一个常数。

3. 缺陷

存在光强度不连续的现象。



7.5.2 Gouraud明暗处理

1. 方法

通过在面片上将光强度进行线性插值来绘制多边形面片。

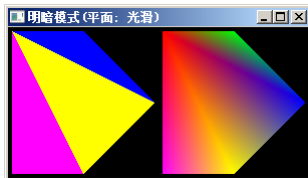
2. 特点

- 强度值沿相邻多边形的公共边界均匀过渡。
- 消除了光强度不连续的现象。

Gouraud明暗处理与平面明暗处理的比较如图7-13所示。

3. 所需计算

- 确定每个顶点处的平均单位法向量。
- 对每个顶点处根据光照模型计算光强度。
- 在多边形投影区域内对顶点光强度进行线性插值。



4. 顶点处平均法向量的计算

如图7-14所示。设面片 S_1, S_2, \dots, S_m 都通过顶点 V ，其单位法向量分别为 $\overrightarrow{N_1}, \overrightarrow{N_2}, \dots, \overrightarrow{N_m}$ ，则顶点 V 处的平均单位法向量为

$$\overrightarrow{N_V} = \frac{\sum_{k=1}^m \overrightarrow{N_k}}{\left| \sum_{k=1}^m \overrightarrow{N_k} \right|}$$

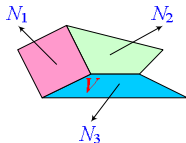


图7-14 平均法向量

5. 沿多边形边界对光强度插值（边界的光强度）

（1）方法。对每一条扫描线，它与多边形交点处的光强度可以通过边界的两端点的光强度进行线性插值得到（只使用扫描线的 y 坐标进行插值）。

（2）举例。如图7-15，若已知某边界两端点 y_1 和 y_2 处的光强度，则沿该边界在扫描线 y 处的光强度为

$$I = I_1 + \frac{I_2 - I_1}{y_2 - y_1}(y - y_1)$$

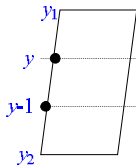


图7-15 沿边界插值

$$\frac{I - I_1}{y - y_1} = \frac{I_2 - I_1}{y_2 - y_1}$$

(3) 使用增量法获得相邻扫描线上后续点的光强度。对于上述例子，沿该边界在扫描线 $y-1$ 处的光强度为

$$\begin{aligned} I' &= I_1 + \frac{I_2 - I_1}{y_2 - y_1}(y - 1 - y_1) \\ &= I - \frac{I_2 - I_1}{y_2 - y_1} \end{aligned}$$

$$I = I_1 + \frac{I_2 - I_1}{y_2 - y_1}(y - y_1)$$

6. 沿扫描线对光强度插值（内部的光强度）

（1）方法。一旦一条扫描线在边界处的光强度已知，内部点的光强度可以通过扫描线在边界处的光强度进行线性插值得到（只使用扫描线的 x 坐标进行插值）。

（2）举例。如图7-16，若已知沿某扫描线在边界 x_1 和 x_2 处的光强度，则沿该扫描线在 x 处的光强度为

$$I = I_1 + \frac{I_2 - I_1}{x_2 - x_1}(x - x_1)$$

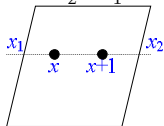


图7-16 沿扫描线插值

$$\frac{I - I_1}{x - x_1} = \frac{I_2 - I_1}{x_2 - x_1}$$

(3) 使用增量法获得沿扫描线后续点的光强度。对于上述例子，则沿该扫描线在 $x+1$ 处的光强度为

$$\begin{aligned} I' &= I_1 + \frac{I_2 - I_1}{x_2 - x_1} (x + 1 - x_1) \\ &= I + \frac{I_2 - I_1}{x_2 - x_1} \end{aligned}$$

$$I = I_1 + \frac{I_2 - I_1}{x_2 - x_1} (x - x_1)$$

7.6[#] 几种高级效果

7.6.1 纹理映射

纹理映射是把图像应用到物体表面的一种技术，类似于在物体表面贴上印花图案。通常将纹理图像中的像素称为纹素，使用 (s, t) 坐标系描述。

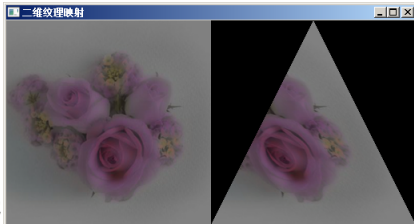
原始的纹理图像一般是一个方形图像，要把它映射到奇形怪状的物体上，一个纹素对应一个屏幕像素一般是做不到的，需要进行纹理映射，如图7-17所示。



图7-17 纹理映射

这里介绍一种非常简单的纹理映射方法。该方法只适用于三角形面片，其他面片需要分解成三角形面片（OpenGL将多边形分解成三角形扇形）。

该方法的思路是首先通过从像素坐标到纹理坐标的变换找到最靠近像素（该像素使用纹理）中心的四个纹素，然后将纹素亮度和像素的当前亮度相结合，计算该像素的结合后亮度，如图7-17所示。结合的方法主要有最近邻插值结合和双线性插值结合等。



1. 像素坐标到纹理坐标的变换

要使用这种映射方法，首先必须构造出从像素坐标到纹理坐标的变换，该变换可以由3个像素坐标-纹理坐标对构造出来。其中，像素坐标-纹理坐标对可通过为物体的每个顶点指定纹理坐标来实现。

由

$$\begin{pmatrix} s_1 & s_2 & s_3 \\ t_1 & t_2 & t_3 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}$$

可得

$$\begin{pmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} s_1 & s_2 & s_3 \\ t_1 & t_2 & t_3 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}^{-1}$$

2. 最近邻插值结合

假设像素 (x, y) 的当前亮度为 f ，需要计算该像素的结合后亮度。计算过程如下。

- 由 (x, y) 获得 (s, t) ，这里的 (s, t) 不一定是一个纹素。
- 令 $s_0 = \text{int}(s)$ ， $t_0 = \text{int}(t)$ ， $s_1 = s_0 + 1$ ， $t_1 = t_0 + 1$ 。显然， $(s_0, t_0), (s_0, t_1), (s_1, t_0), (s_1, t_1)$ 是纹素。
- 在 $(s_0, t_0), (s_0, t_1), (s_1, t_0), (s_1, t_1)$ 中选取距离 (s, t) 最近的纹素 (u, v) ，将 (u, v) 处的亮度 g 与像素的亮度 f 结合。如图7-18左侧所示。

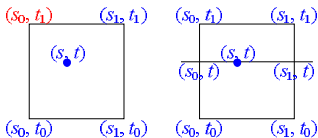


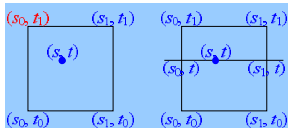
图7-18 最近邻插值结合与双线性插值结合

3. 双线性插值结合

假设像素 (x, y) 的当前亮度为 f ，需要计算该像素的结合后亮度。双线性插值结合方法的前2个步骤与最近邻插值结合方法完全相同。

- 设 $(s_0, t_0), (s_0, t_1), (s_1, t_0), (s_1, t_1)$ 处的亮度分别是 $g_{00}, g_{01}, g_{10}, g_{11}$ 。
- 用 $g_0 = \frac{g_{01} - g_{00}}{t_1 - t_0}(t - t_0) + g_{00}$ 计算 (s_0, t) 处的亮度 g_0 。
- 用 $g_1 = \frac{g_{11} - g_{10}}{t_1 - t_0}(t - t_0) + g_{10}$ 计算 (s_1, t) 处的亮度 g_1 。
- 用 $g = \frac{g_1 - g_0}{s_1 - s_0}(s - s_0) + g_0$ 计算 (s, t) 处的亮度 g 。
- 将 (s, t) 处的亮度 g 与像素的亮度 f 结合。如图7-18右侧所示。

$$\frac{y - y_0}{x - x_0} = \frac{y_1 - y_0}{x_1 - x_0} \Rightarrow y = \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) + y_0$$



7.6.2 雾化

1. 关于雾化

计算机图像有时候由于过于清晰和锐利，反而显得不太逼真。我们可以通过添加雾效果，使整幅图像变得更加逼真。所谓雾效果，就是使远处的物体看上去逐渐变得模糊。

2. 雾化模型

一般的雾化模型是使用一个雾化因子将雾的颜色与物体的实际颜色进行结合。设物体的实际颜色为 c_r ，雾的颜色为 c_f ，融合因子为 f ，则结合后的颜色为 $fc_r + (1 - f)c_f$ 。

3. 雾化因子的计算

雾化因子通常通过雾离观察点的距离计算，计算雾化因子的方程通常包括线性雾化方程、指数雾化方程和二次指数雾化方程等。

(1) 线性雾化方程。设雾化的起始距离为 z_s ，结束距离为 z_e ，雾离观察点的距离为 z ，则雾化因子 $f = (z_e - z) / (z_e - z_s)$ 。若 $f < 0$ ，则规定 $f = 0$ ，若 $f > 1$ ，则规定 $f = 1$ 。

(2) 指数雾化方程。设雾的浓度 ρ ，雾离观察点的距离为 z ，则雾化因子 $f = e^{-\rho z}$ 。若 $f > 1$ ，则规定 $f = 1$ 。

(3) 二次指数雾化方程。设雾的浓度 ρ ，雾离观察点的距离为 z ，则雾化因子 $f = e^{-(\rho z)^2}$ ①。若 $f > 1$ ，则规定 $f = 1$ 。

① 此处，OpenGL 手册和 MSDN 等资源的描述有误。

7.7 练习题

7.7.1 基础知识题

1. 请概括深度缓冲器算法的步骤。
2. 对于光照模型 $I = k_a I_a + k_d I_l (\overline{N} \cdot \overline{L}) + k_s I_l (\overline{N} \cdot \overline{H})^{n_s}$ ，公式中的三项分别表示何含义？公式中各符号的含义是什么？
3. 如何计算几个法向量的平均单位法向量。
4. 简要介绍一些处理透明物体表面的方法（至少一种）。
5. 描述沿多边形边界对光强度插值的增量方法。
6. 描述沿扫描线对光强度插值的增量方法。

7.7.2 阶段实习题

编制一个线框多面体绘制程序，要求能够显示规范化观察体中任何凸多面体的线框模型，其中隐藏线使用虚线显示，可见线使用实线显示。