零碎知识点整理

- 第一章——概述
- 第二章
  - 多边形表示
  - 字符
  - 反走样
  - 消隐面算法的优缺点比较
- 几何造型技术
  - 三种描述物体的三维模型
  - 八叉树的优缺点
  - 三角网格
  - 网格处理
  - 网格简化——层次细节网格简化
  - 网格细分
  - 特征敏感网格重剖P120
- 第四章——真实感知图形学
  - 常见的颜色模型
    - RGB
    - CMY颜色模型
    - HSV模型
    - HSV和RGB的联系
  - Phong光照明模型P133
  - 增量式光照模型
    - Gourand明暗处理模型——双线性光强插值P135
    - Phong明暗处理——双线性法向插值P136
    - 两种算法的对比
    - 两种算法的缺点
  - 阴影的生成
    - 阴影生成的算法
  - 光线跟踪模型P150
    - **光线跟踪算法伪代码描述**⚠
    - 最基本的运算——求交点运算P152
    - 如何加速

零碎知识点整理

第一章——概述

1. 图像和图形

> 图像是指计算机内以位图形式存在的灰度信息
>
> 图形有明确的的几何信息，强调几何表示，由场景的几何模型和景物的物理属性共同组成

2. 关于发展史

> Steven A.Coons提出了超限插值的曲面造型新思想，通过插值四条任意的边界曲线构造曲面
>
> 计算机图形学的最高奖是以Coons的名字命名的
>
> 1970年 Bouknight提出了第一个光反射模型
>
> Gourand提出了漫反射模型——插值的思想，即Gourand明暗处理
>
> Phong提出了简单光照明模型，即Phong模型
>
> Whitted提出了第一个光投射模型，即Whitted模型
>
> 图形学最权威的会议——ACM SIGGRAPH

第二章

多边形表示

1. 多边形的两种表示方式

> 1. 顶点表示
>
>   使用多边形的顶点序列表示多边形
>
>   表示直观，几何意义明显，占内存小，易于进行几何变换
>
>   但是没有明确的指出那些像素在多边形内部，不能直接用于面着色
>
> 2. 点阵表示
>
>   使用位于多边形内部的像素集合刻画多边形
>
>   丢失了许多几何信息
>
>   便于帧缓冲器表示图形，是面着色需要的图形的表示形式

2. 扫描转化

   将多边形的顶点表示转化为点阵表示称为多边形的扫描转换

字符

字符有两种表示方式，分别是点阵字符和矢量字符

1. 点阵字符

   每一个字符由一个位图表示，字位为1，表示字符的笔画经过此位置，对应于此位置的像素颜色为字符颜色；字位为0，表示字符的笔画不经过此位置，对应此位置的像素置为背景色

> 如何显示
>
> 1. 首先从字符库中将他的位图检索出来
> 2. 将检索到的位图写到帧缓冲器中

> 如何变换
>
> 需要对表示字符位图中的每一像素进行

2. 矢量字符

   记录笔画的信息而不是整个位图，具有存储空间小，美观，变换方便等优点

> 如何显示
>
> 1. 首先从字符库中检索出来字符的信息
> 2. 取出端点坐标，进行适当的几何变换，在根据端点信息显示字符

> 如何进行旋转，变换等
>
> 只需要对笔画端点进行变换

字符裁剪的三个精度：串精度，字符精度，笔画或像素精度

反走样

走样的定义：使用离散的量表示连续的量所引起的失真现象称为走样

反走样的定义：用于减小或消除走样的技术称为反走样

反走样常用的方法：

> 1. 提高分辨率（只能减轻不能消除锯齿问题）
> 2. 区域采样
>   1. 反走样的效果较好
>   2. 有时可以划分为多个子像素，计算中心点落在直线段内子像素的个数，用于估计面积的近似值
>
>   最后将屏幕中像素的亮度置为最大灰度值乘以相交区域的面积近似值
>
> 1. 像素的亮度与相交区域的面积成正比，而与相交区域落在像素的位置无关，仍会产生锯齿效应
> 2. 直线条上沿着理想直线方向上的相邻两个像素有时会有较大的灰度差
>
> 3. 加权区域采样

消隐面算法的优缺点比较

几何造型技术

三种描述物体的三维模型

> 1. 线框模型
>
>   最早用于表示物体的模型
>
>   使用顶点和棱边来表示物体
>
>   由于没有面的信息，因此不能明确的定义点和物体之间的关系
>
>   不能处理消隐图，明暗色图等问题
> 2. 曲面模型
>
>   在线框模型的基础上增加了物体面的信息，使用面的集合来表示物体，使用环来定义面的边界
>
>   可以用于实现面面求交，线面消隐，明暗色图
>
>   无法分析物体的整体性质，如表面积，体积，重心
> 3. 实体模型
>
>   最高级模型，可以完整的表示物体的所有形状信息
>
>   可以无歧义的确定一个点在物体的内部还是外部

八叉树的优缺点

> 优点
>
> 1. 形体表示的数据结构简单
> 2. 简化了形体的集合运算，对形体执行交，并，差运算时只需要同时遍历参加集合运算的两个形体相应的八叉树，无需进行复杂的求交运算
> 3. 简化了隐藏线（或面）的消除，因为在八叉树的表示中，形体上的各元素已安札空间形成了一定的顺序
> 4. 分析算法适合并行处理

> 缺点
>
> 1. 占用的存储空间多
> 2. 只能那个近似的表示形体，不以获取形体的边界信息

三角网格

1. 三角网格的优点

> 1. 容易通过三维扫描技术大量获取
> 2. 采用足够多的面片可以以任意精度逼近复杂的曲面
> 3. 网格模型的数据结构简单，光照计算和显示速度快且适合硬件并行处理

2. 三角网格的半边结构P115

> 半边结构也称为双向链接边表，将一条无向的边拆分为两条有向的半边，半边的方向总是沿着逆时针方向

> 每一个三角形都存储三条半边
>
> 每一个半边需要存储
>
> 1. 该半边的原顶点origin(e)
> 2. 与该半边在同一三角形中的下一条边next(e)
> 3. 与该半边同处于一条边的对边opposite(e)
> 4. 该半边所属的面incFace(e)

网格处理

> 简化:使用较少的面片来表示几何，提高绘制的效率，通常会有一定的损失
>
> 减小网格的大小，便于存储和运输
>
> 降低网格的复杂度
>
> 提高绘制速度

> 细分：以原始的网格为基础，按照一定的规则生成包含更多面片的几何

> 重剖：为了获取更规则的网格模型，模型可能会有更多或更少的面片

> 光顺：为了得到与原网格基本一致的模型，但是更光滑，从而去除不需要的几何细节或噪声

网格简化——层次细节网格简化

定义：在不影响画面视觉效果的条件下，通过逐次简化景物的表面细节来减少场景的几何复杂性，从而提高绘制算法的效率

基本方法P118

> 1. 顶点删除操作
> 2. 边压缩
> 3. 面片收缩

如何实现P119

> 1. 首先定义每一次操作后网格带来的误差，并且用这个误差作为原始网格上每一个基本元素的权值，插入到一个按权值递增排序的队列中
> 2. 对网格进行循环基本化简操作
> 3. 每一次循环中选取队首权值最小的操作并执行，之后更新变化的网格信息，重新计算改变了的网格的基本元素的误差
> 4. 重新插入到有序队列中，再开始下一个循环，知道队列的最小误差达到用户设定的阈值或者得到预期的化简数目

网格细分

LOOP细分P120

1. 只适合三角网格

2. 先增加顶点的数目，之后更新每一个顶点的位置
3. 每一次操作之后，三角形的数目变成原来个数的4倍

特征敏感网格重剖P120

第四章——真实感知图形学

1. 可见光的波长为380-780（或者400-700）mm，**单位是mm**
2. 在图形学中一般采用三基色颜色系统
3. 颜色的三个特性

> 从心里学和视觉的角度
>
> 1. 色调（HUE）
> 2. 饱和度（saturation）
> 3. 亮度（lightness）
>
> 从光学的角度
>
> 1. 主波长（domainant wavelength）
> 2. 纯度（purity）
> 3. 明度（Luminance）

4. 三色学说：人眼的视网膜中存在三种椎体细胞，包含不听的色素，对光的吸收和反射特性不同，对于不同的光就有不同的颜色感觉

   三色学说是真实感知图形学的生理基础，是颜色视觉中最基础，最根本的理论，是RGB等其他颜色模型的基础

5. CIE色度图P129
6. 任何颜色 模型都是可见光的一个子集

常见的颜色模型

RGB

1. RGB颜色模型就是三维直角坐标颜色系统的一个单位正方体
2. 用途是在某个颜色域内方便的指定颜色
3. 常用与显示设备，例如彩色阴极射线管等彩色光栅图形显示设备

   加性原色

CMY颜色模型

1. 以红，绿，蓝的补色——青色，品红，黄为原色构成的CMY颜色模型，常用与从白光中过滤某种颜色，即**减性原色系统**
2. 面型硬件，**打印设备**

HSV模型

面向用户

HSV和RGB的联系

P131

RGB主对角线对应于HSV空间的V轴

Phong光照明模型P133

模拟物体表现对光的反射作用

1. 模型假设

> 1. 光源假定位点光源
> 2. 反射作用被细分为镜面反射和漫反射
> 3. 只考虑物体对直接光照的反射作用，物体之间的光反射作用使用环境光统一表示

2. 模型的问题

> 1. Phong模型提显示出的物体像塑料，没有质感
> 2. 环境光只是常量，没有考虑物体之间相互的反射光
> 3. 镜面反射的颜色是光源的颜色，与物体的材质无关
> 4. 镜面反射的计算在入射角很大时会产生失真
> 5. 是经验模型，与真实的物理模型存在偏差

增量式光照模型

实现多边形之间光滑的过度，使连续的多边形呈现均匀的光强分布

> 基本思想
>
> 1. 在每一个多边形的顶点处计算合适的光照明强度或其他参数
> 2. 在各个多边形的内部进行均匀的插值
> 3. 最后得到多边形的光滑颜色分布

Gourand明暗处理模型——双线性光强插值P135

1. 算法描述

> 1. 计算多边形顶点的平均法向
> 2. 使用Phong光照明模型计算顶点的平均光强
> 3. 插值计算离散边上的各点的光强
> 4. 插值计算多边形内部各点的光强

Phong明暗处理——双线性法向插值P136

1. 算法描述

> 1. 保留双线性插值，对多边形上的点和内域中的各点都采用增量法
> 2. 对顶点的法向量进行插值，而顶点的法向量使用相邻多边形的法向量的平均值
> 3. 由插值得到的法向量，计算每一个像素的光亮度
> 4. 假定光源与视点在无穷远处，光强只是法向量的函数

两种算法的对比

1. 双线性光强插值能够有效的显示漫反射效果，且计算量小
2. 双线性法向插值可以产生正确的高光区域，但是计算量大

两种算法的缺点

1. 两种模型得到的物体变圆轮廓是折线而不是光滑曲线
2. 由于透视的原因，使等间距扫描线产生不均匀的效果
3. 插值效果决定于插值的方向，不同的插值方向会得到不同的插值结果

阴影的生成

1. 阴影生成的条件

> 1. 眼睛可以观察到，但是光线照射不到
> 2. 附着在物体表面或者地板上

阴影生成的算法

1. 采用透视投影变换，首先在以光源为坐标原点的世界空间中，为每一个像素打上标记，目的是区分场景中的每一个物体对光源是否可见
2. 同样的，再回到相机空间（人眼观测），遍历每一个像素，判断每一个像素对于人眼是否可见
3. 如果像素可以被相机观测到但是不能被光源照射到，且附着在物体表面，则该像素应该生成阴影，需要在计算光强使人为的修改kd,ks,ks等系数

（从渲染管线的角度，也可以为根据像素对于人眼和光源的可见情况，标注不同的材质，每一种材质定义了一组kd,ks,ka等系数，用于计算光强）

光线跟踪模型P150

1. 算法描述P150

2. 光线跟踪算法的递归终止条件

> 1. 该光线未碰到任何物体
> 2. 光线碰到了背景，光源，或光线跑出了视景体
> 3. 光线经过多次反射和折射后，能量衰减到小于一定的阈值
> 4. 光线反射或者折射的次数大于一定的阈值，即光线跟踪的次数大于一定的阈值

光线跟踪算法伪代码描述⚠

```
void RayTracing(strat,direction,weight,color){
    if(weight<Min_weight){
        color=black;
    }
    else{
        计算光线与所有物体的交点中距离起点start最近的点;
        if(没有交点){
            color=black;
        }
        else{
            I_loacl=在交点处使用局部光照模型计算出的光强;
            计算反射方向R;
            RayTracing(最近的交点,R,weight*w_r,I_r);  //考虑光线的衰减
            计算折射方向;
            RayTracing(最近的交点,T,weight*w_t,I_t); //考虑光线的衰减，需要乘以衰减系数
            color=I_local+K_s*I_r+K_t*I_t;
        }
    }
}
```

最基本的运算——求交点运算P152

如何加速

1. 自适应深度控制（如上代码）
2. 层次包围盒技术
3. 八叉树