

《计算机图形学基础》模拟试题答案(一) 答案

一、问答题 (25 分, 每题 5 分)

1、简述 Cohen-Sutherland 裁剪方法的思想, 并指出与之相比, 中点裁剪方法的改进之处, 及这种改进的理由。

答: Cohen-Sutherland 裁剪算法的思想是: 对于每条线段 P_1P_2 分为三种情况处理。(1) 若 P_1P_2 完全在窗口内, 则显示该线段 P_1P_2 简称“取”之。(2) 若 P_1P_2 明显在窗口外, 则丢弃该线段, 简称“弃”之。(3) 若线段既不满足“取”的条件, 也不满足“弃”的条件, 则求线段与窗口交点, 在交点处把线段分为两段。其中一段完全在窗口外, 可弃之。然后对另一段重复上述处理。

中点分割算法的大意是, 与 Cohen-Sutherland 算法一样首先对线段端点进行编码, 并把线段与窗口的关系分为三种情况: 全在、完全不在和线段和窗口有交。对前两种情况, 进行同样的处理。对于第三种情况, 用中点分割的方法求出线段与窗口的交点。即从 P_0 点出发找出距 P_0 最近的可见点 A 和从 P_1 点出发找出距 P_1 最近的可见点 B, 两个可见点之间的连线即为线段 P_0P_1 的可见部分。从 P_0 出发找最近可见点采用中点分割方法: 先求出 P_0P_1 的中点 P_m , 若 P_0P_m 不是显然不可见的, 并且 P_0P_1 在窗口中有可见部分, 则距 P_0 最近的可见点一定落在 P_0P_m 上, 所以用 P_0P_m 代替 P_0P_1 ; 否则取 P_mP_1 代替 P_0P_1 。再对新的 P_0P_1 求中点 P_m 。重复上述过程, 直到 P_mP_1 长度小于给定的控制常数为止, 此时 P_m 收敛于交点。

改进之处在于, 对第三种情况, 不直接解方程组求交, 而是采用二分法收搜索交点。

这种改进的理由是: 计算机屏幕的像素通常为 1024×1024 , 最多十次二分搜索即可倒像素级, 必然找到交点。而且中点法的主要计算过程只用到加法和除 2 运算, 效率高, 也适合硬件实现。

2、在 Phong 模型 $I = I_a K_a + I_p K_d (L \cdot N) + I_p K_s (R \cdot V)^n$ 中, 三项分别表示何含义? 公式中的各个符号的含义指什么?

答: 三项分别代表环境光、漫反射光和镜面反射光。 I_a 为环境光的反射光强, I_p 为理想漫反射光强, K_a 为物体对环境光的反射系数, K_d 为漫反射系数, K_s 为镜面反射系数, n 为高光指数, L 为光线方向, N 为法线方向, V 为视线方向, R 为光线的反射方向。

3、针对多面体模型,直接用 Phong 模型绘制会有什么问题? 简述两种增量式光照模型(明暗处理)的基本思想, 并指出两个算法的主要区别。

答: 针对多面体模型, 使用 Phong 模型绘制会在多边形与多边形之交界处产生明暗的不连续变化, 影响了曲面的显示效果, 即马赫带效应。如果增加多边形个数, 减小每个多边形的面积, 当然也能改善显示效果。但是这样一来, 数据结构将迅速膨胀, 导致操作的空间与时间上升。增量式光照模型的基本思想是在每一个多边形的顶点处计算合适的光照明强度或法向量, 然后在各个多边形内部进行均匀插值, 得到多边形光滑的颜色分布。它包含两个主要的算法: 双线性光强插值和双线性法向插值, 又被分别称为 Gouraud 明暗处理和 Phong 明暗处理。两种算法的主要区别为: Gouraud 明暗处理采用光强插值, 而 Phong 明暗处理采用法向插值。

4、解释走样和反走样的概念, 并给出三种以上反走样方法(只写名称)。

答: 在光栅显示器上显示图形时, 直线段或图形边界或多或少会呈锯齿状。原因是图形信号是连续的, 而在光栅显示系统中, 用来表示图形的却是一个个离散的象素。这种用离散量表示连续量引起的失真现象称之为走样; 用于减少或消除这种效果的技术称为反走样。反走样的方法有: 提高分辨率、区域采样和加权区域采样。

5、依次写出用中点画线法进行直线扫描转换, 从点(5, 5)到(15, 11)经过的象素点。

解:

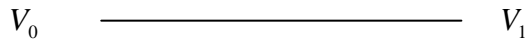
5	5
6	6
7	6
8	7
9	7
10	8
11	9
12	9
13	10
14	10
15	11

二、选择题 (25 分, 每题 5 分)

6、提出“计算机图形学”的一些基本概念和技术, 确定了计算机图形学作为一个崭新科学分支的独立地位, 从而被称为图形学之父的是: a。

- a. Ivan E. Sutherland b. Pierre Bézier c. Steven A. Coons d. Bui-Tuong Phong

7、函数 $\Phi(t) = \begin{cases} V_0 + \frac{V_1 - V_0}{3}t, & 0 \leq t \leq 1 \\ V_0 + \frac{V_1 - V_0}{3} + (t-1)\frac{2(V_1 - V_0)}{3}, & 1 \leq t \leq 2 \end{cases}$ 的图形为



在 $t=1$ 处的连续性为 abcd

- a. C^0 b. G^0 c. C^1 d. G^1 e. C^∞

8、在 Warnack 消隐算法中, 窗口与多边形的关系包括哪些 abcd ?

- a. 内含 b. 相交 c. 包围 d. 分离

9、一条以 P_0, P_1, P_2, P_3, P_4 为控制顶点的 4 阶(三次)B 样条曲线, 其节点向量为

$\{0, 0, 0, 1, 2, 3, 4, 4, 4\}$, 则其定义域为: c

- a. (0,4) b. (1,2) c. (1,3) d. (1,4)

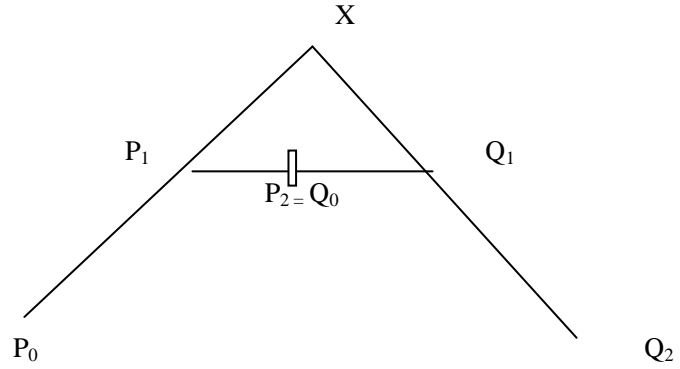
10、光线跟踪算法的主要计算量在于: d

- a. 基于 Phong 模型的明暗度计算 b. 反射方向计算 c. 折射方向计算 d. 求交计算

三 (10 分)、设一条二次 Bezier 曲线的控制顶点为 P_0, P_1, P_2 , 另一条二次 Bezier 曲线的控制顶点为 $Q_0, Q_1, Q_2, P_2 = Q_0$, 写出两条曲线可以精确合并 (表示) 为一条二次 Bezier 曲线的条件。

解: 如下图所示, 由于可以精确合并, 说明两曲线是由一条曲线在参数 $0 < \lambda < 1$ 处分割而来, 如下图所示, 假设原曲线的控制顶点为 P_0, X, Q_2 . 由 de Casteljau 算法, 有:

1. 首先要求 $P_1, P_2(Q_0), Q_1$ 三点共线



$$2. \frac{Q_2 - Q_1}{Q_1 - X} = \frac{Q_1 - Q_0}{P_2 - P_1} = \frac{X - P_1}{P_1 - P_0},$$

于是有：

$$Q_1 - \frac{Q_2 - Q_1}{Q_1 - Q_0} (P_2 - P_1) = P_1 + \frac{P_1 - P_0}{P_2 - P_1} (Q_1 - Q_0)$$

四（10分）、设一条三次 Bezier 曲线的控制顶点为 P_0, P_1, P_2, P_3 ，对曲线上一点 $P\left(\frac{1}{2}\right)$ ，

及一个给定的目标点 T ，给出一种调整 Bezier 曲线形状的方法，使得 $P\left(\frac{1}{2}\right)$ 精确通过点 T 。

解：假设我们改变其中的一个控制顶点，比如将 P_1 调整到 $P_1 + \lambda$ ，使得使得 $P\left(\frac{1}{2}\right)$ 精确通

过点 T ，改变后的曲线记为 $\hat{P}(t)$ 则有：

$$\hat{P}(t)\Big|_{t=\frac{1}{2}} = \sum_{i=0}^3 P_i B_{i,3}(t)\Big|_{t=\frac{1}{2}} + \lambda \cdot B_{1,3}\left(\frac{1}{2}\right)$$

即

$$T = P\left(\frac{1}{2}\right) + \lambda \cdot B_{1,3}\left(\frac{1}{2}\right)$$

所以，只需将 P_1 调整到 $P_1 + \left(T - P\left(\frac{1}{2}\right)\right) / B_{1,3}\left(\frac{1}{2}\right)$ ，即可。

五（10分）、写出只用点 Z-Buffer（一个变量）的消隐算法，比较其区别和传统 Z-Buffer（一个二维数组）的消隐算法的区别。

答：只用点 Z-Buffer（一个变量）的消隐算法如下：

```
Z-Buffer()
{
    帧缓存全置为背景色
    //扫描整个屏幕
    for(屏幕上的每个象素(i,j))
    {
        深度缓存变量 zb 置最小值 MinValue
        for(多面体上的每个多边形 Pk)
        {
            if(象素点(i,j)在 pk的投影多边形之内)
            {
                计算 Pk在(i,j)处的深度值 depth;
                if(depth 大于 zb)
                {
                    zb = depth;
                    indexp = k;
                }
            }
        }
        if(zb != MinValue) 计算多边形 Pindexp在交点 (I,j) 处的光照颜色并显示
    }
}
```

传统的 Z-Buffer 算法需要有帧缓存和一个二维数组的 z 深度缓存。在把显示对象的每个面上每一点的属性（颜色或灰度）值填入帧缓冲器相应单元前，要把这点的 z 坐标值和 z 缓冲器中相应单元的值进行比较。是遍历多边形；而点 Z-buffer 算法是遍历象素，再判有哪些多边形的投影包含此象素，作深度比较。但是在不用开二维数组的 Z-buffer 的同时，带来了缺点是：反复进行点在多边形内部的判断，使算法效率降低。

六（10分）、1. 写出光线跟踪递归函数的伪代码。

2. 描述光线跟踪加速的层次包围盒方法。

解：

1.

```
RayTracing(start, direction, weight, color)
{
    if ( weight < MinWeight )
```

```

    color = black;
else
{  计算光线与所有物体的交点中离 start 最近的点;
  if( 没有交点 )
    color = black;
  else
  {
     $I_{local}$  = 在交点处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*  $w_r$ ,  $I_r$ );
    计算折射方向  $T$ ;
    RayTracing(最近的交点,  $T$ , weight*  $w_t$ ,  $I_t$ );

    color =  $I_{local} + K_s I_r + K_t I_t$ ;
  }
}
}

```

2. 包围盒技术的基本思想是用一些形状简单的包围盒(如球面、长方体等)将复杂景物包围起来, 求交的光线首先跟包围盒进行求交测试, 若相交, 则光线再与景物求交, 否则光线与景物必无交。它是利用形状简单的包围盒与光线求交的速度较快来提高算法的效率的。简单的包围盒技术效率并不高, 因为被跟踪的光线必须与场景中每一个景物的包围盒进行求交测试。包围盒技术的一个重要改进是引进层次结构, 其基本原理是根据景物的分布情况, 将相距较近的景物组成一组局部场景, 相邻各组又组成更大的组, 这样, 将整个景物空间组织成树状的层次结构。进行求交测试的光线, 首先进入该层次的根节点, 并从根节点开始, 从上向下与各相关节点的包围盒进行求交测试。若一节点的包围盒与光线有交, 则光线将递归地与其子节点进行求交测试, 否则, 该节点的所有景物均与光线无交, 该节点的子树无需作求交测试。

七 (10 分)、已知 4 阶 B 样条曲线的节点矢量为 $0, 0, 0, 0, 0.5, 1, 1, 1, 1$,

$$\text{de Boor 递推公式为: } d_j^l = \begin{cases} d_j & l=0 \\ (1-\alpha_j^l)d_{j-1}^{l-1} + \alpha_j^l d_j^{l-1} & l=1 \cdots k \end{cases}, \quad \alpha_j^l = \frac{t-t_j}{t_{j+k+1-l}-t_j}$$

以下是 $t=0.4$ 处求值的 de Boor 三角形, 请补齐空的。

