《计算机图形学基础》模拟试题(二)答案

一、问答题 （25 分，每题 5 分）

1、 列举三种常见的颜色模型，简要说明其原理和特点。

答：所谓颜色模型就是指某个三维颜色空间中的一个可见光子集，它包含某个颜色域的所有颜色。常用的颜色模型有 RGB、CMY、HSV 等。

RGB 颜色模型通常用于彩色阴极射线管等彩色光栅图形显示设备中，它是我们使用最多、最熟悉的颜色模型。它采用三维直角坐标系，红、绿、蓝为原色，各个原色混合在一起可以产生复合色。

CMY 颜色模型以红、绿、蓝的补色青（Cyan）、品红（Magenta）、黄（Yellow）为原色构成，常用于从白光中滤去某种颜色，又被称为减性原色系统。印刷行业中基本使用 CMY 颜色模型。

HSV（Hue，Saturation，Value）颜色模型是面向用户的，对应于画家的配色方法。

2、 列举三种以上常见的曲面、曲面求交方法。

答：曲面与曲面求交的基本方法有代数方法、几何方法、离散方法和跟踪方法四种。

代数方法：代数方法利用代数运算，特别是求解代数方程的方法求出曲面的交线。

几何方法：几何方法是利用几何的理论，对参与求交的曲面的形状大小、相互位置以及方向等进行计算和判断，识别出交线的形状和类型，从而可精确求出交线。对于交线退化或者相切的情形，用几何方法求交可以更加迅速和可靠。

离散方法：离散方法求交是利用分割的方法，将曲线不断离散成较小的曲面片，直到每一子曲面片均可用比较简单的面片，如四边形或者三角形平面片来逼近，然后用这些简单面片求交得到一系列交线段，连接这些交线段即得到精确交线的近似结果。

跟踪方法：跟踪方法求交是通过先求出初始交点，然后从已知的初始交点出发，相继跟踪计算出下一交点，从而求出整条交线的方法。

3、 给出四次 Bezier 曲线退化为三次 Bezier 曲线，控制顶点 P_0, P_1, P_2, P_3, P_4 应满足的条件。

答：退化条件是将曲线展开成幂级数形式后，所有 t^4 的系数只和为零，即

$$\sum_{i=0}^{4} \Delta^i P_0 = 0 \quad 或 \quad \sum_{i=0}^{4} P_i C_4^i (-1)^{4-i} = 0$$

4、 列举三种形体表示的常见方法。

答：分解表示、构造表示和边界表示。

5、 计算机图形学的概念是谁在其博士论文中提出的？

答：Ivan E. Sutherland。

二、选择题 (25 分，每题 5 分)

6、ACM Siggraph 最高奖是以_____c_____的名字命名的。

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

7、中点法扫描转换以(0，0)， (5，2)为端点的直线段时，不经过下面哪个点__c__？

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

8、五个控制顶点的三次 B 样条的节点向量应该由几个节点构成__d__？

a. 5　　b. 7　　c. 8　　d. 9

9、多项式 Bezier 曲线不能表示哪种几何元素__b c__？

a. 直线　　b. 圆弧　　c. 双曲线　　d. 抛物线

10、属于空间剖分技术的光线跟踪加速方法有：__a c__

a. 三维 DDA　　b. 层次包围盒　　c. 八叉树　　d. 自适应深度控制

三 (10 分)、给定型值点(0,0),(0,100),(100,0),(100,100)，如对应的参数为$0,\frac{1}{3},\frac{2}{3},1$，反求插值这四个型值点的三次 Bezier 曲线的控制点。

答：假设控制顶点为b_0, b_1, b_2, b_3，由 Bezier 曲线的公式，将参数为$0,\frac{1}{3},\frac{2}{3},1$代入曲线方程，即有：

$b_0 = (0,0)$，

$b_3 = (100,100)$，

$$(0,100) = \frac{8}{27}b_0 + \frac{4}{9}b_1 + \frac{2}{9}b_2 + \frac{1}{27}b_3$$

$$(100,0) = \frac{1}{27}b_0 + \frac{2}{9}b_1 + \frac{4}{9}b_2 + \frac{8}{27}b_3$$

解方程组可得 $b_1 = (-\frac{350}{3}, \frac{1000}{3}), b_2 = (\frac{650}{3}, -\frac{700}{3})$。

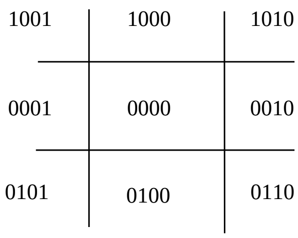
四 (10 分)、描述 Cohen-Sutherland 裁剪算法。

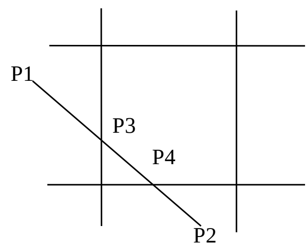
答：该算法的思想是：对于每条线段 P_1P_2 分为三种情况处理。（1）若 P_1P_2 完全在窗口内，则显示该线段 P_1P_2 简称“取”之。（2）若 P_1P_2 明显在窗口外，则丢弃该线段，简称“弃”之。（3）若线段既不满足“取”的条件，也不满足“弃”的条件，则在交点处把线段分为两段。其中一段完全在窗口外，可弃之。然后对另一段重复上述处理。

为使计算机能够快速判断一条直线段与窗口属何种关系，采用如下编码方法。如下图，延长窗口的边，将二维平面分成九个区域。每个区域赋予 4 位编码 CtCbCrCl.其中各位编码的定义如下：

$$C_t = \begin{cases} 1 & y > y_{\max} \\ 0 & other \end{cases} \quad C_b = \begin{cases} 1 & y < y_{\min} \\ 0 & other \end{cases} \quad C_r = \begin{cases} 1 & x > x_{\max} \\ 0 & other \end{cases} \quad C_l = \begin{cases} 1 & x > x_{\min} \\ 0 & other \end{cases}$$

裁剪一条线段时，先求出 P_1P_2 所在的区号 code1,code2。若 code1=0,且 code2=0，则线段 P_1P_2 在窗口内，应取之。若按位与运算 code1&code2≠0，则说明两个端点同在窗口的上





方、下方、左方或右方。可判断线段完全在窗口外，可弃之。否则，按第三种情况处理。求出线段与窗口某边的交点，在交点处把线段一分为二，其中必有一段在窗口外，可弃之。在对另一段重复上述处理。在实现本算法时，不必把线段与每条窗口边界依次求交，只要按顺序检测到端点的编码不为 0，才把线段与对应的窗口边界求交。

五 (10 分)、 (1) 推导 Beizer 曲线的升阶公式。

(2) 给定三次 Beizer 曲线的控制顶点(0,0),(0,100),(100,0),(100,100)，计算升阶一次后的控制顶点。

解：

设给定原始控制顶点 $P_0, P_1, \cdots, P_n$，定义了一条 n 次 Bezier 曲线：

$$P(t) = \sum_{i=0}^{n} P_i B_{i,n}(t) \qquad t \in [0,1]$$

增加一个顶点后，仍定义同一条曲线的新控制顶点为 $P_0^*, P_1^*, \cdots, P_{n+1}^*$，则有：

$$\sum_{i=0}^{n} C_n^i P_i t^i (1-t)^{n-i} = \sum_{i=0}^{n+1} C_{n+1}^i P_i^* t^i (1-t)^{n-i}$$

对上式左边乘以 $(t+(1-t))$，得到：

$$\sum_{i=0}^{n} C_n^i P_i t^i (1-t)^{n+1-i} + t^{i+1}(1-t)^{n-i}) = \sum C_{n+1}^i P_i^* t^i (1-t)^{n+1-i}$$

比较等式两边 $t^i (1-t)^{n+1-i}$ 项的系数，得到：

$$P_i^* C_{n+1}^i = P_i C_n^i + P_{n-1} C_n^{j-1}$$

化简即得：

$$P_i^* = \frac{i}{n+1} P_{i-1} + \left(1 - \frac{i}{n+1}\right) P_i \qquad (i = 0,1,\cdots,n+1)$$

其中 $P_{-1} = P_{n+1} = 0$。

升阶一次后的控制顶点为(0,0)，(0,75)，(50,50)，(100,25)，(100,100)。

六(10 分)、用 de Boor 算法，求以(30,0),(60,10),(80,30),(90,60),(90,90)为控制顶点、以 T=(0,0,0,0,0.5,1,1,1,1)为节点向量的的三次 B 样条曲线在 t=1/4 处的值。

解：由 de Boor 算法， $P(t) = \sum_{i=j-k+1}^{j} P_i N_{i,k}(t) = \sum_{i=j-k+2}^{j} P_i^{[1]}(t) N_{i,k-1}(t)$，

按公式：

有以下的 de Boor 三角形：

(30,0)

(60,10) → (45,5)

(80,30) → (65,15) → (55,10)

(90,60) → (82.5,37.5) → (69.375,20.625) → (62.1875,15.3125)

(90,90)

即 B 样条曲线在 t = 1/4 处的值为（62.1875,15.3125）

七 (10 分)、描述多边形扫描转换的扫描线算法。

答：算法的思想：

扫描线多边形扫描转换算法是按扫描线顺序，计算扫描线与多边形的相交区间，再用要求的颜色显示这些区间的象素，即完成填充工作。区间的端点可以通过计算扫描线与多边形边界线的交点获得。对于一条扫描线，多边形的填充过程可以分为四个步骤：

（1）求交：计算扫描线与多边形各边的交点；

（2）排序：把所有交点按 x 值递增顺序排序；

（3）配对：第一个与第二个，第三个与第四个等等；每对交点代表扫描线与多边形的一个相交区间，

（4）填色：把相交区间内的象素置成多边形颜色，把相交区间外的象素置成背景色。

但求交会导致算法效率低，为了能很好地利用图形地连贯性，我们可以引进多边形表、活性边表等数据结构，并利用增量算法避免求交运算。具体算法如下：

算法过程

```
void polyfill (POLYGON polygon, int color)
{
    for(各条扫描线 i )
    {
        初始化新边表头指针 NET [i];
        把 ymin = i  的边放进边表 NET [i];
    }
    y =  最低扫描线号;
    初始化活性边表 AET 为空;
    for (各条扫描线 i)
    {
        把新边表 NET[i]中的边结点用插入排序法插入 AET 表，使之按 x 坐标递增顺
```

序排列;

遍历AET表，把配对交点区间(左闭右开)上的象素(x, y)，用drawpixel(x, y, color)改写象素颜色值;

遍历 AET 表，把 ymax = i 的结点从 AET 表中删除，并把 ymax > i 结点的 x 值递增 Dx;

若允许多边形的边自相交，则用冒泡排序法对 AET 表重新排序;

}

} /* polyfill */