《计算机图形学》实习指导书

一、上机实验的目的和要求

通过实验，使学生了解计算机生成图形的方法与原理，掌握用计算机生成图形的一般技术，为后面的专业知识的学习奠定一定的基础。另一方面，通过实践加强学生对所学知识的理解与掌握，进一步培养学生实际动手能力。

由于实习时间有限，因此要求学生：在实习前事先对实验内容及方法非常熟悉，尽量做到先写出实验算法与过程，再上机调试运行，最终解决理论与实践的矛盾。

二、上机实验的具体安排与内容

实习时间安排为：第 6、9、12、15 周星期共四次，具体内容为：

1. 基本图形（直线）的模拟生成[数值微分法（DDA）、Bresenham 法]
2. 曲线的拟和（分段 Bezier 曲线）或插值（三次 Spine 曲线）
3. 线段的窗口裁剪（编码法）或多边形裁剪（Sutherland 法）
4. 二维几何变换（文件读取、缩放、平移、比例变换等）

三、实验指导——流程图

实验一、图形函数的使用及模拟直线的生成（DDA 法或 Bresenham 法）

1. 图形函数的运用

图形库的打开：

头文件　#include “graphics.h”

图形初始化：　initgraph(&dr, &mode, “path”);

图形函数：　putpixel(int x, int y, int color);

setcolor(int color);　…

关闭图形模式：　closegraph();

图形操作举例：

```
#include “graphics.h”
main()
{ int dr = DETECT, mode;
 initgraph(&dr, &mode, “”);
 setcolor( RED ) ;
 …
 closegraph();
}
```

2. 打点方式模拟生成直线段

通过打点的方式生成直线段，使学生了解图形显示的基本原理。

a. 数值微分法（Digital Differential Analysis: DDA）

其工作原理是增量法。计算机中生成直线时，通过一个个栅格表示所通过的像素点，其有一定的大小，因此，可以通过所求直线在某一个方向上增加一个单位，而另一个方向每次只有不到一个单位 k 的增量，从而可以实现从起点绘制直线段到终点。

（ DDA 直线模拟生成流程图 ）

输入端点坐标（x_s, y_s）, (x_e, y_e)
计算其 x,y 增量 dx,dy
计算 e = 1.0/max(|dx|,|dy|), 及 addx, addy addx = e*dx,　　addy = e*dy
x = x_s+0.5,　y = y_s +0.5
for(i = 0; i<= max(|dx|,|dy|) ; i ++) 　　x = x + addx 　　y = y + addy 　　putpixel(x, y, color)

b. Bresenham 方法（ |K| <= 1 ）

其工作原理是误差决定走步。该方法在计算过程中，每迭代一次，直线在绝对值最大方向走一步，另一方向是否走一步由走最大方向产生的误差决定。若误差大于 0.5，走一步并且误差进行改正，否则继续在绝对值最大方向走步。

(Bresenham 法直线模拟生成流程图 |k|<=1)

读入直线端点坐标（x_s, y_s）, (x_e, y_e)
计算增量 dx,dy，　及其 k = dy/dx
判别函数先赋初值e=-0.5, x=x_s, y=y_s

for(i = 0; i <= | dx |; i ++)
　　putpixel(x, y, color);
　　x ++;
　　e = e + k;
　　e >= 0
　　T　　F
　　y ++;
　　e = e –1;

实验二、逼近曲线 Bezier 曲线或三次 Spline 曲线的生成

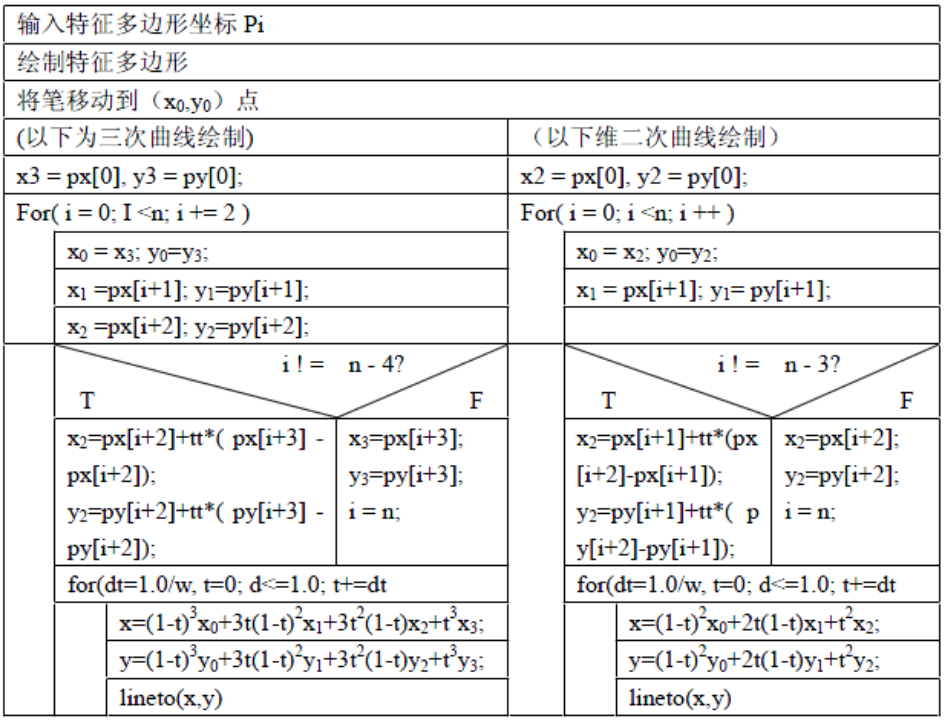
学习了直线段的模拟算法，通过给定的观测点，通过程序生成连续的光滑曲线。这里主要实践逼近曲线 Bezier 曲线的二次曲线和三次曲线。另外，个别同学可以在此实践完成后，实现三次 Spline 曲线的绘制。Bezier 曲线为这样一条曲线，其是由一组折线集，或称为 Bezier 特征多边形来定义的，曲线的起点和终点与该多边形的起点、终点重合，且多边形第一条边和最后一条边表示了曲线在起点和终点处的切矢量方向。曲线形状区域多边形形状，其次数严格依赖于决定该曲线的点数。方程表达式为：

$C(t) = \sum P_i \cdot B_{i,n}(t) \quad (0 \leqslant t \leqslant 1)$

其中：　$B_{i,n}(t) = (n!/(i!(n-i)!)) \cdot t^i \cdot (1-t)^{n-i}$

算法实现中，通过一次 Bezier 曲线内插的方式得到分段曲线的连接点。

（ 二次、三次 Bezier 曲线绘制流程图 ）

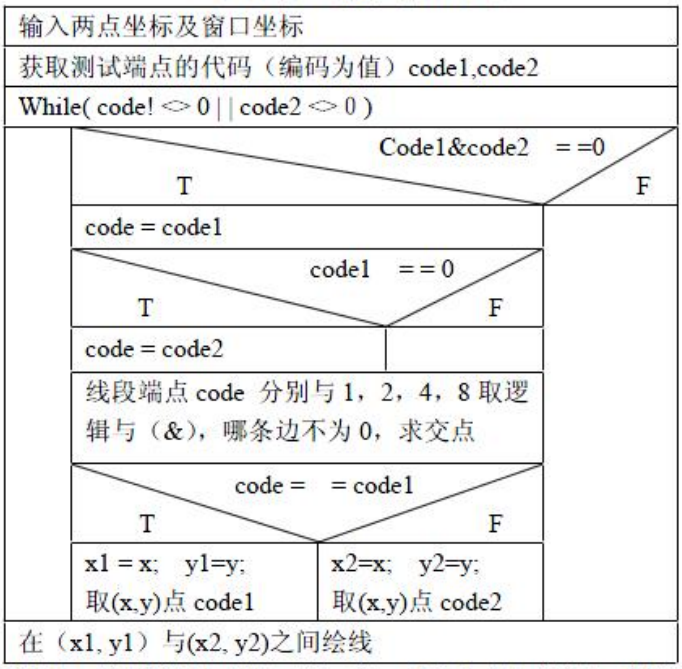


这里仅介绍 Bezier 曲线的流程图，三次 Spline 曲线程序，可以依据《计算机图形学》（第三版，清华大学出版社）编写。

实验三、线段的窗口裁剪（编码法）或多边形裁剪（Sutherland 法）

了解分区编码法对图形区域的分区原则与分类原则，体会在图形裁剪中分区方法的优点。

（矩形窗口裁剪线段流程图）



详细算法见：《计算机图形学》（第三版，清华大学出版社）或其他相关书籍

实验四、二维几何变换（文件读取、缩放、平移、比例变换等）

1. 文件的读取与操作

基本过程：

头文件： #include “stdio.h”

文件指针： FILE * fp;

文件打开： fp = fopen(“filename”, “r”);

文件读取：fscanf(fp,”%s”,str);

文件关闭：fclose(fjp);

举例：

```
#include “stdio.h”
main()
{ FILE *fp;
 if((fp = fopen（“filename”,”r”）) = = NULL )
 { printf(“Can’t open this file!\n”); exit(0); }
  while( !feof( fp ))
  { fscanf(fp,”%s”,str);
  …
  …}
  fclose(fp);
  }
```

2. 二维基本变换

读取文件，对读入坐标进行平移、缩放、比例等基本变换，实现图形的显示。

（对图形的平移、缩放、比例等变换流程图）

<table>
<tr><td colspan="2">打开文件读取图形区域的范围（minx, miny）(maxx, maxy)</td></tr>
<tr><td colspan="2">根据图形窗口大小，进行基本视见变换</td></tr>
<tr><td colspan="2">打开图形模式及相关设置</td></tr>
<tr><td colspan="2">While(！feof (fp)) 当文件还未读取到文件结尾时</td></tr>
<tr><td rowspan="4"></td><td>读取目标的坐标及编码</td></tr>
<tr><td>依据编码对目标进行分类</td></tr>
<tr><td>对坐标进行仿射变换（平移、缩放、比例等变换）</td></tr>
<tr><td>绘图输出</td></tr>
<tr><td colspan="2">关闭图形窗口及已打开文件</td></tr>
</table>

cn.txt 文件格式

-5.548866 -3.671638
5.521260 6.183811
2222 2222
11111 11111
2.895496 -0.064799
2.923685 0.003252
2.931383 0.015953
2.947298 0.044392
2.960657 0.074075
3333 3333
11111 11111
-1.966496 0.571291
-1.959647 0.588467
-1.948304 0.623942
-1.946272 0.653638
-1.955313 0.682640
-1.954082 0.722878
-1.921709 0.760547
1111 1111
11111 11111
3.168472 1.850717
3.148548 1.837824
3.127174 1.824312
3.113370 1.816465
3.077149 1.802963
3.034641 1.787575
2.983339 1.765583
2.931399 1.723962
4444 4444
11111 11111
-2.007579 -0.154429
11111 11111
0.129720 -0.191492