计算机应用丛书

计算机图像技术及其应用

姚家奕　等编著

国防工业出版社

计算机应用丛书

计算机图像技术及其应用

姚家奕　姜万录　李朝晖　编著
姚传胤　审校

国防工业出版社
·北京·

图书在版编目(CIP)数据

计算机图像技术及其应用/姚家奕等编著. —北京:国防工业出版社,1998.9
(计算机应用丛书)
ISBN 7-118-01867-8

Ⅰ.计… Ⅱ.姚… Ⅲ.计算机图形学 Ⅳ.TP391.4

中国版本图书馆 CIP 数据核字(98)第 00238 号

国防工业出版社出版发行
(北京市海淀区紫竹院南路 23 号)
(邮政编码 100044)
北京怀柔新华印刷厂印刷
新华书店经售

开本 787×1092 1/16 印张 21¼ 485 千字
1998 年 9 月第 1 版 1998 年 9 月北京第 1 次印刷
印数:1—4500 册 定价:29.00 元

前　言

计算机图像技术是伴随计算机的发展而迅速发展起来的一门新兴学科。它包括计算机图形学和计算机图像处理。计算机图像处理与计算机图形学的结合，已成为计算机辅助设计(CAD)、计算机辅助制造(CAM)、计算机辅助教学(CAI)等的重要基础，目前它广泛地应用于遥感技术、工业控制、生物医学工程以及文化教育、军事、公安等各个领域。

计算机多媒体技术的发展，使计算机图像技术的应用更加广阔，对其在各个领域中的应用一定会起到巨大的推动作用。

为适应这些新技术发展的需要，我们编写了此书。全书共十三章。详细地介绍了使用Turbo C图形功能编程的方法，介绍了图像变换、图像处理、计算机绘图、动画显示等原理与方法。各章的主要内容是：

第一章　显示模式。介绍字符显示模式与图形显示模式的设置方法，存储模式与混合模式的编程方法。

第二章　字符模式的字符输出。介绍在字符显示模式条件下设置字符颜色、字符背景色、字符亮度、字符闪烁，以及字符屏幕操作等。

第三章　图形模式的彩色控制。介绍各种调色板，如CGA、EGA、VGA等的选择方法，并介绍在编程中绘图颜色、背景色和图形填充的设置方法。

第四章　图形函数。介绍Turbo C图形库中画像素点、画线段、画圆弧、画扇形、画椭圆、画多边形等函数的使用方法。

第五章　图形模式字符输出。介绍在图形屏幕模式时输出字符的方法。如设置字形的大小、显示方向以及设置字符宽度、高度和字体等。

第六章　二维图像的变换与显示。介绍二维图像的平移、旋转、缩放等算法。并介绍变换算法的矩阵表示及齐次坐标变换和复合变换算法。

第七章　三维图像的变换与显示。介绍三维图像的变换方法及复合变换。其中还介绍三维图像的透视图的坐标变换和曲面的显示方法。

第八章　三维图像的拟合与消隐。介绍曲线拟合的抛物线插值、拉格朗日插值多项式、Bezier曲线、B样条曲线、曲面拟合的三次参数样条的混合函数表示、双三次Coons曲面的数学描述以及三维图像的消隐处理等。

第九章　计算机图像处理。介绍图像信息的去本底及归一化运算、图像的平滑处理、锐化处理、快速平滑算法以及像素的插值方法等。

第十章　计算机动画显示。介绍用视见区产生动画的方法以及用异或写方式、用屏幕操作函数、用设置图形页等方法实现动画。

第十一章　图像处理软件开发的实用技巧。介绍320×200，256色VGA13H图形模式的开发，支持鼠标器程序，汉字处理及显示的方法等。

第十二章　典型程序设计例。介绍图像变换、图像信息处理、汉字显示以及动画显示

等典型的程序设计例。

第十三章　系统安装程序设计。介绍程序系统的设计方法。

上述各章可划分为下面几部分：第一部分，第一章到第五章为计算机绘图，使用 Turbo C 编译工具编程显示图形；第二部分，第六章到第八章为图像的变换与拟合等，第三部分，第九章为计算机图像处理；第四部分，第十章为计算机动画显示；第五部分，第十一章至第十三章为图像处理软件开发的实用技巧及应用例。

在编写本书的过程中，参考了国内各种教材及资料，并引用了国外的有关资料。姚家奕负责主编，编写第二章至第七章及第九章和第十章中§10.3和§10.4节。姜万录编写第一章和第十二章。李朝晖编写第八章、第九章中§9.4节的五和§9.5节、第十章、第十一章和第十三章。张淑清和杨军参加了部分编写工作。姚传胤负责全书的审校工作。

本书在编写过程中得到了北方交通大学的大力支持，并得到秦皇岛渤海铝业公司王新利同志的大力协助。在此一并表示衷心感谢。

由于编者水平所限，书中不妥之处在所难免，恳请广大读者批评指正。

编　者

目　录

第一章　显示模式

计算机的屏幕显示有两种方式，字符显示方式和图形显示方式，分别称为字符显示模式与图形显示模式。字符显示模式主要用于字符的显示输出，而图形显示模式则主要用于图形的显示输出。在这一章我们介绍这两种显示模式的使用方法。首先介绍字符显示模式。

§1.1　字符显示模式

字符显示模式是用于显示字符的，它决定屏幕上显示的字符的行列数。此时屏幕上可访问和显示的最小单位为一个字符。

一、显示字符

在字符显示模式下只有字符才能显示出来。它在屏幕上的位置由它们的行与列数决定，而且规定，左上角定位为 1.1。

例如在 C80 模式下屏幕分成 80×25＝2000 个单元，列为 1 至 80，行为 1 至 25，如图 1-1 所示。

各单元由一个字符和一种属性组成。

字符：可以是任意的 ASCII 码字符，或扩充字符。

属性：用于控制字符的显示色、背景色和是否闪烁。

上述字符和属性可以分别控制。

目前 Turbo C 可支持五种以上字符显示模式，列于表 1-1。

表 1-1　IBM 系列微机显示模式

编　号	(符号值)	显示模式	分辨率	图形卡
0	(BW40)	字符、黑白	40×25	CGA、EGA
1	(C40)	字符、16 种色	40×25	CGA、EGA
2	(BW80)	字符、黑白	80×25	CGA、EGA
3	(C80)	字符、16 色	80×25	CGA、EGA
7	(MONO)	字符、黑白	80×25	MDA
−1	(LAST)	启用原字符模式		

二、显示模式控制

对上述各种字符显示模式、各种显示属性，Turbo C 都提供了相应的函数加以控制。在使用之前对它们都要事先设置。使用字符屏幕方式时，先要设置字符显示模式。

其调用函数为

Void textmode (int mode)

其中，mode 为显示模式：只能选用字符模式 BW40、BW80、C80、LAST 等。

调用此函数时需要标题文件 conio.h。

如调用 textmode()，显示屏幕将被清屏，属性采用常规属性 C80；若 mode 为 LAST 时，则重新选择前面使用过的字符模式。这种情况一般是用于从图形模式回到字符模式。

例如：

```
# include<conio.h>
main()
{
    textmode(C80);
    ⋮
}
```

在此程序中即将屏幕置为 80 列×25 行的彩色字符方式，可实现字符的彩色显示，显示 2000 个字符。

在未经定义的情况下，大多数的计算机系统都使用 80 列字符模式 3。

三、显示窗口

设置了显示模式之后，要输出字符时还要确定输出的区域，通常这一区域是整个屏幕。但 Turbo C 还允许在屏幕上划出一个矩形区域，作为字符输出的范围。

这一个限定的输出范围称为一个窗口。一旦设置了窗口，则以后所有的输出只影响该窗口的内部，而不影响屏幕的其它部分。

这个设置窗口的函数是

```
void window (int left,int top,int right,int bottom);
```

调用此函数所需的标题文件是<conio.h>

其中各个参数 left、top、right、bottom 分别表示窗口的第一列、第一行，最后一列、最后一行。窗口的显示范围如图 1-2 所示。

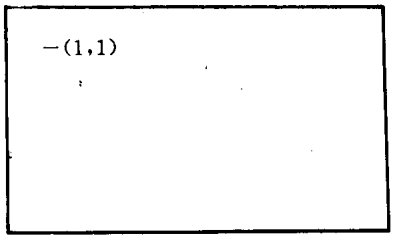


图 1-1　字符显示模式

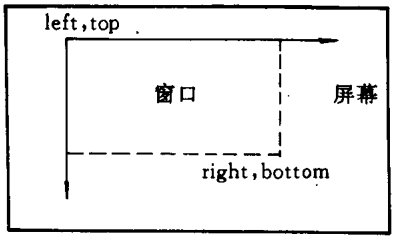


图 1-2　窗口显示范围

窗口的最重要特性是，对设置了窗口的输出可以防止向窗口以外的溢出，即如果某些输出超出了窗口的边界，那么只有窗口内部的部分才能显示出来，而超出的那些部分则自动被剪裁掉。被截去的部分不被认为是出错。

一旦窗口函数调用成功，则所有定位坐标都是相对窗口的，而不是整个屏幕。此时要使用窗口的字符输入输出函数。

常用的窗口字符输入输出函数有以下几种：

cprintf()：格式化输出到当前窗口。

cputs()：输出一个字符串到当前窗口。

putch()：输出单个字符到当前窗口。

getche()：读一个字符并回显到当前窗口。

cgets()：读一个字符串并回显到当前窗口。

上面所有窗口函数均需 conio. h 标题文件。这些函数不允许把字符写到当前窗口以外。而 getch()函数则不接收来自当前窗口以外的输入，在超出窗口的位置从键盘输入的部分则不能读入。

当使用缺省窗口显示(全屏幕)时，无论是用窗口的基本 I/O 函数，还是使用标准 I/O 函数，其作用均相同。但是，应注意的是，标准 I/O 函数不适用窗口屏幕环境。

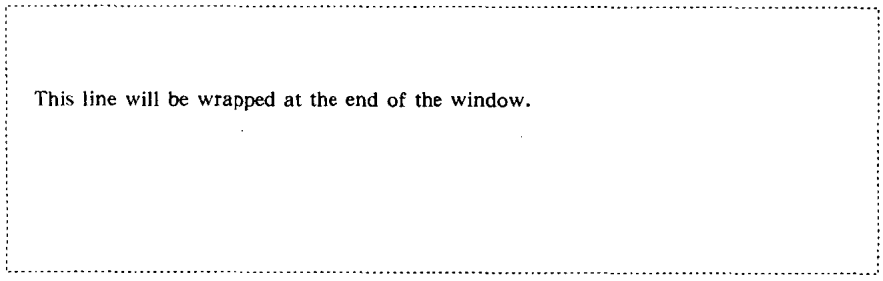
下面我们来分析程序例 test1-1. c 的输出。

程序例 test1-1. c

test1-1. c

```
#include<conio. h>
void border (int,int,int,int);
main()
{
    clrscr();

    window(3,2,40,9);
    border(3,2,40,9);
    gotoxy(3,3);
    cputs("This line will be wrapped at the end of the window.");
    getche();
}

void border (startx,starty,endx,endy)
int startx,starty,endx,endy;
{
    register int i;

    gotoxy (1,1);
    for (i=0;i<=endx-startx;i++)
        putch('-');

    gotoxy(1,endy-starty);
    for (i=0;i<=endx-startx;i++)
        putch('-');

    for(i=2;i<endy-starty;i++)
```

```
    {
        gotoxy(1,i);
        putch('|');
        gotoxy (endx-startx+1,i);
        putch('|');
    }
}
C>test1-1 ↵
```



test1-1 图

此程序显示如上所示的输出。首先建立一个窗口 window(3,2,40,9),然后由 border(3,2,40,9)函数绘出此窗口的边界。再调用函数 cputs()写一行比窗口长的字符串,超出窗口的部分则自动移到窗口内的下一行。

程序中未设置显示模式,系统则自动定义为 C80 模式。

在此程序中使用了 gotoxy()函数,此函数的原型为

```
void gotoxy (int x,int y);
```

它将字符屏幕上的光标移动到相对窗口的 x、y 所指定的位置上。其中参数 x、y 分别表示输出位置在当前窗口中的列和行数。

因此

```
gotoxy(3,3);
cputs(……);
```

则表示在相对窗口的 x=3、y=3 处开始写一行字符串。

border()函数是自定义函数,该函数画出窗口的边框。它首先将光标移至(1,1)处,然后由 putch('—')函数按 for(i=0;i<=endx-startx;i++)循环,画上边框。再将光标移至(1,endy-starty)处后,由 putch('—')函数按 for(i=0;i<=endx-startx)循环,画下边框。最后由第三个循环

```
for(i=2;i<endy-starty;i++)
```

画出左、右两边的竖线。

程序在开始时使用了清屏函数

```
void clrscr(void)
```

该函数清除当前字符窗口中的字符,清除从当前光标位置到该行结束的所有字符,但光标位置保持不变。所用标题文件为 conio. h。

§1.2 图形显示模式

一般在未经定义时计算机系统都工作在80列字符模式3(C80),此时Turbo C的图形函数是不能工作的。如果要想在图形模式下工作,则首先要设置某一图形模式。为此要使用初始化函数

```
void far initgraph(int far * driver,int far * mode, char far * path);
```

标题文件为graphics.h。

此函数用于初始化计算机系统。所谓初始化,即是将图形驱动软件装入内存。为了实现各种图形的输出,对应不同的图形卡配备有不同的图形驱动程序,需要将它们调入内存。

下面我们来分别介绍初始化的几个参数。

一、driver(图形卡)

参数driver决定图形卡及其所对应的驱动软件。例如要选择EGA图形卡,我们可取

```
driver=EGA;
```

表1-2列出了Turbo C支持的各种图形模式。

表1-2 Turbo C支持的图形模式

图形卡(符号值)	图形模式(符号值)	分辨率	颜色数
DETECT(0)			
CGA(1)	CGAC0(0)	320×200	4
	CGAC1(1)	320×200	4
	CGAC2(2)	320×200	4
	CGAC3(3)	320×200	4
	CGAHI(4)	640×200	2
MCGA(2)	MCGAC0(0)	320×200	4
	MCGAC1(1)	320×200	4
	MCGAC2(2)	320×200	4
	MCGAC3(3)	320×200	4
	MCGAMED(4)	640×200	2
	MCGAHI(5)	640×480	2
EGA(3)	EGALO(0)	640×200	16
	EGAHI(1)	640×350	16
EGA64(4)	EGA64LO(0)	640×200	16
	EGA64HI(1)	640×350	4
EGAMONO(5)	EGAMONOHI(3)	640×350	2
IBM8514(6)	IBM8514LO(0)	640×480	
	IBM8514HI(1)	1024×760	
HERCMONO(7)	HERCMOHONOHI(0)	720×348	2

（续）

图形卡(符号值)	图形模式(符号值)	分辨率	颜色数
ATT400(8)	ATT400C0(0)	320×200	4
	ATT400C1(1)	320×200	4
	ATT400C2(2)	320×200	4
	ATT400C3(3)	320×200	4
	ATT400CMED(4)	640×200	2
	ATT400CHI(5)	640×400	2
VGA(9)	VGALO(0)	640×200	16
	VGAMED(1)	640×350	16
	VGAHI(2)	640×480	16
PC3270(10)	PC3270HI(0)	720×350	2

二、**mode**(图形模式)

图形模式的选择由参数 mode 决定，它决定图形的分辨率。

在字符模式下可访问的最小单位是一个字符；而在图形模式状态下屏幕上可访问的最小单位则是一个像素，它是图形显示器上可访问的最小点。可访问的最小点数愈多，则表示分辨率也愈高，图形就愈清晰。

例如在 EGA 卡的 EGAHI 模式时分辨率为 640×350，共 224000 个像素。可用如下程序语句实现

```
driver=EGA;
mode=EGAHI;
```

应指出的是，在图形模式时屏幕的左上角定为 0,0，如图 1-3 所示。

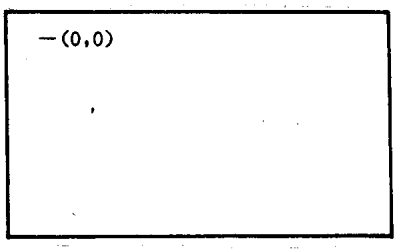


图 1-3　图形显示模式

此外，图形模式中 DETECT(0)表示，初始化函数 initgraph()将自动在当前系统下查找显示器的图形卡类型，并选用最高分辨率的屏幕显示模式。

三、**path**(**路径名**)

path 为查找图形设备驱动程序所用的路径名。

Turbo C 图形系统包含两部分：一部分是与机器无关的图形库 Graphics. LIB(其标题文件为 graphics .h)；另一部分是与具体图形显示卡有关的图形(设备)驱动程序，它以文件的形式存入磁盘，其扩展名为 BGI(Borland 图形接口)。例如：

EGA 卡的设备驱动程序为 EGA. BGI；

VGA 卡的设备驱动程序为 VGA. BGI；

CGA 卡的设备驱动程序为 CGA. BGI。

因此使用 Turbo C 图形函数时一定要有图形库 graphics. LIB，并将其存在机器内；另外还要将相应的图形设备驱动程序调入内存。

路径名 path 即指出 BGI 文件所在的位置。如果 BGI 文件在当前盘内，那么，路径名可省略，可用空格代替，如用" "代替。如果在当前盘的子目录中存放有 BGI 文件，路径名

可用"a:bgi"或"b:bgi"表示。

下面是一个使用初始化函数的简单程序例(程序例 test1-2.c)。

程序例 test1-2.c

test1-2.c

```
#include"graphics.h"
main()
{
    int driver,mode;
    driver=DETECT;

    initgraph(&driver,&mode,"");
    printf("driver=%d,mode=%d",driver,mode);

    getch();
    closegraph();
}
```

C>test1-2 ↵

driver=9,mode=2

此程序可用于测试系统当前设备的最高图形卡和图形模式。DETECT 查找屏幕配备的图形卡,并选用最高的图形模式。

在系统的 graphics.h 文件中将图形卡和模式均定义为符号值,例如 1 代表 EGA,可看成是整型数。因此程序 test1-2.c 的输出结果为

driver=VGA,mod=VGAHI

在配有 EGA 卡的系统中,如果要将屏幕置为高分辨率模式,可使用下面程序段

```
#include<graphics.h>

main()
{
    int driver,mode;

    driver=EGA;
    mode=EGAHI;
    initgraph(&driver,&mode,"");
        .
        .
        .
    closegraph();
    printf("this is not in graphics.");
}
```