

在普通彩色打印机上打印精美的彩色图像

叶长青 韩宁

摘要:在普通 24 针彩色打印机上逼真地打印彩色图像的方法,并用汇编语言编写了打印屏幕图像的程序。

目前,大多数计算机都配备了高分辨率的图形显示器,并配有高级图形卡 VGA,可在屏幕上经常见到许多颜色丰富、层次感强、精美逼真的彩色图像,还可以用彩色扫描仪输入 256 色的彩色图像。能否在普通 24 针只有黑、红、蓝、黄四种颜色的打印机上逼真地输出这些彩色图像呢?在打印机上形成图像的方法与报纸杂志上印刷图片的方法基本相同,如果我们近距离仔细看一幅彩报图片,就会发现它是由许多深浅不同的红、蓝、黄颜色小点组成的。事实上自然界中的各种各样的颜色都可以分解成不同成分的红、蓝、黄三原色,绘画中的调色、彩色电视、彩色相片等都是利用彩色的三原色原理,用三种原色合成各种各样的颜色,逼真地反映自然界中的彩色。打印机也是用点阵方式打印图像的,对于每一种颜色的点,在一小区域中用点的疏密来仿真该区域的色度,通过不同色度的三原色迭加,就比较逼真地反映出原来图像的颜色了。

本文通过 VGA 的方式 13H 的屏幕硬拷贝程序的编制过程来说明怎样打印 256 色图像的方法。

整个处理过程为:首先将彩色图像的调色板即 VGA 的 DAC 读出,根据调色板加色三原色的亮度值,换算成打印机减色三原色的色度值,再根据打印机减色三原色的色度,对每种都进行变换,变为相应的疏密点阵,最后在打印机上输出点阵,达到打印彩色图像的目的。下面分步讨论。

一、读 VGA 的 DAC 并换算成疏密点阵

每当按下屏幕拷贝键,就进入中断 05H 的中断处理程序,中断 05H 负责拷贝屏幕的具体操作,首先是保护中断前的现场,即各个寄存器的值,判断当前屏幕方式是否是方式 13H,即 256 色屏幕方式,如果不是就转到原来中断 05H 的处理程序来拷贝其它显示方式的屏幕,这就进入打印

256 色图形方式的处理工作。

首先是怎样得到现在屏幕方式的调色板,即各个点的红、绿、蓝三种加色的三原色亮度值。为此,我们先看看 VGA 在 256 种颜色下是如何工作的。对于 VGA 显示器来说,它事实上可以显示的颜色远远超过 256 种,方式 13H 下,同一屏幕上最多显示 256 色,这和 VGA 的工作方式有关。在方式 13H 下,显示的图形存放在显示缓冲区中,缓冲区从微机的 A000:0 地址开始,屏幕上的每个点在显示缓冲区中占一个字节,共 8 个二进制位,所以最多同时能显示 2 的 8 次方种颜色,即 256 色。在显示缓冲区中存放的并不是真正颜色的三原色亮度值,而存放的是颜色在调色板寄存器的寄存器号。VGA 有 256 个调色寄存器,每个寄存器有 18 位,分成三组,每组 6 位,分别表示红、绿、蓝三原色的亮度值,而 6 位二进制值能够表示 2 的 6 次方即 64 种不同亮度,所以每个调色板寄存器能选择 $64 \times 64 \times 64 = 262144$ 种颜色的任意一种,而 VGA 只有 256 个调色板寄存器,每个都能选择 262144 种颜色的一种,所以,VGA 在同一屏幕上能够出现 262144 种颜色的任意 256 种。每个彩色图像,在方式 13H 下显示,需要一个调色板寄存器表和每点的颜色号,即对应的调色板寄存器号,所有点的颜色号,依次存放在 A000:0 屏幕显示区中。不同图像的调色板寄存器表是不同的,因此,每次拷贝屏幕时,都要读出调色板寄存器,进行三原色的点阵的疏密换算。

读 DAC 寄存器的方法是用 VGA 的 BIOS 提供的中断 10H 的 10H 号功能调用的 17H 号子功能调用实现的,中断前参数如下:

AH=10H	功能号
AL=17H	子功能号
BX=0	要读的 DAC 寄存器块开始的寄存器号
CX=256	要读的 DAC 寄存器个数
ES:DX	读出 DAC 时存放数据的起始地址

中断返回后,ES:DX 中为 DAC 寄存器的值,每三个字节为

一组,分别是红、绿、蓝的亮度值,总共 256 组。

由于显示器显示彩色的原理与彩色电视机的原理是一样的,合成彩色是由红、绿、蓝三种原色加色实现的,例如红光、绿光、蓝光比例一样时混合起来就是白色,不同亮度的相同比例三种原色混合就是不同亮度的白色,即不同灰度的黑白颜色。又如,不同亮度相同成份的红、绿二色组成的是不同亮度的黄色;红、蓝二色混合为品红;蓝、绿二色混合为天蓝。而打印机打印彩色和绘画、照片、彩色印刷原理一样,是根据减色原理,用品红、天蓝、黄色三原色合成的。例如,相同比例的品红、天蓝、黄色混合一起是黑色,而不是白色。相同比例的天蓝、黄色混合一起是绿色;品红、天蓝混合为蓝色;品红、黄色混合为红色。因此,加色用的三原色与减色法用的三原色是不同的。对于同一种颜色的加色法与减色法合成颜色时虽然三原色不同,但每种原色的色度和亮度是有一定关系的。规律如下:

原色	加色法亮度	原色	减色法色度
红色	R	天蓝	NOT R
绿色	G	品红	NOT G
蓝色	B	黄色	NOT B

根据这个规律,把从 VGA 的 DAC 寄存器表中的值逐个取反,得到减色法的三原色色度值。这样得出的色度为 64 级灰度,但是打印机放大倍数小则不能实现 64 级灰度,然后再实行线性变换,变为能够实现的级别。最后利用点阵块中可打印点数的多少来模拟色度,从打印机输出。

二、打印图像

本文以 CR3240 打印机为例,来说明打印彩色图像的过程。

1. 打印机图像模式的选择

从理论上讲,印刷或打印时,应该选用尽可能细密的点阵,以使输出的图像尽可能地与显示的图像相像。因此应选择有较高点阵密度的图像打印模式。CR3240 打印机的 24 针打印图像模式有 11 种,其最高密度模式为 6 倍密度模式,水平方向为 360 点/英寸,垂直方向为 180 点/英寸。虽然 CR3240 打印机支持 1/360 英寸换行量,由于打印头针距为 1/180 英寸,故采用错针套打方法,可以实现在水平和垂直方向点阵密度都为 360 点/英寸。但是,由于打印针有一定的直径,如果点密度选的过密,必然引起打印过程不能可靠地打印出密度值较小和密度值较大的区域的灰度阴影。在密度较低时,白色区域变的太小,

以致于被四周的彩色点填满。因此最好选择 3 倍密度的图像打印模式。

2. 打印机控制指令

不同型号的打印机的指令是不同的,因此,对不同打印机应该使用不同的指令,下面介绍 CR3240 打印机本文将用到的指令。

ESC 3 n 设定 n / 180 英寸换行量。

本文在没有打印完所有色时,中间换色打印时临时换行时取 n=0;在打印完一行真正换行时取 n=24,即换行量等于 24 个针距,保证了打印机打印的各行图像能够较好的联结起来。

ESC r n 选择打印机颜色。

- n=0 黑色
- n=1 品红
- n=2 天蓝
- n=3 蓝色
- n=4 黄色
- n=5 红色
- n=6 绿色
- n=7 黑色

本文只使用 n=0 黑色、n=1 品红、n=2 天蓝、n=4 黄色。打印时应该由浅入深逐色打印,首先打印黄色,其次打印品红色,再次打印天蓝色,最后打印黑色。

ECS * m nL nH data.....data 设定打印机图像模式,并发送图像数据

本文选择 3 倍密度图像打印方式,方式数 m=39,一行打印总列数 n=nL+256×nH,data 为每列打印 3 字节的图像数据。

由于 CR3240 打印机是 24 针打印机,每行打印时,打印的每列为 24 个点,而打印机与计算机通讯中使用 8 位信息,因此所有打印列都需要 3 个字节的图像数据。将一个垂直列划分为 3 个各有 8 只针的部分,分别考虑,第一字节对应一列图像的上边 8 针,第二字节对应该列中间的 8 针,第三字节对应该列下面的 8 针,每一部分的 8 个针与一字节 8 位二进制数据对应关系为:

·	最高位	128
·		64
·		32
·		16
·		8
·		4
·		2
·	最低位	1

打印时,哪个针需要打印,对应该位为 1,否则为 0。

3.放大倍数与点对应的点阵选择

VGA 显示器在方式 13H 下,为 320×200 点,纵横比为 200/320;而标准 VGA 方式 12H 为 640×480 点,在方式 12H 下纵向点距和横向点距是相等的,纵横比应该 480/640;因此方式 13H 横向点距与纵向点距比例为:

$$\frac{(200/320) \quad 200 \quad 640}{(480/640) \quad 320 \quad 480} = \times = 400/480 = 5/6$$

因此,为了使打印出来的图像纵横比例一致,我们选择横向放大倍数为 5 倍,纵向放大倍数为 6。因此,屏幕上一,对应打印机的 6×5 点阵。

三、程序实现

本文彩色拷贝屏幕程序用汇编语言实现,程序如下:

```

;CSPRTSCR.ASM
;彩色屏幕硬拷贝程序
;生成COM 程序方法
;MASM CSPRTSCR;
;LINK CSPRTSCR;
;EXE2BIN CSPRTSCR CSPRTSCR.COM

CODE SEGMENT
    ASSUME CS:CODE
    ORG 100H

START: JMP ZK
KPFLAGDB 0 ;拷贝标志
YSB1 DB 256 DUP (0) ;黑色灰度表
YSB DB 256 DUP (0,0,0) ;DAC表
ROM17 DD 0F000EFD2H ;原中断17H入口
ROM05 DD ? ;原来中断05入口
YSHD1 DB 0,0,0,0,0,0,0,0,0,0,88H,88H,0CCH,0EEH,0FFH,0FFH
DB 0,0,44H,44H,66H,66H,66H,77H,77H,77H,77H,0FFH,0FFH
DB 0FFH,0FFH,0FFH
DB 0,22H,0,66H,66H,66H,66H,66H,77H,77H,77H,77H,77H
DB 77H,77H,0FFH
DB 0,0,0,0,0,44H,66H,66H,66H,77H,77H,77H,77H,77H
DB 77H,0FFH
;下面为点对应点阵灰度密度数据
YSHD2 DB 0,1,92H,92H,92H,92H,92H,92H,0B6H,0B6H,0B6H,0B6H
DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
DB 0,0,0,0,0,49H,49H,49H,49H,49H,49H,49H
DB 49H,49H,49H,49H,0DBH,0DBH,0FFH,0FFH
DB 0,0,0,0,24H,24H,24H,24H,24H,24H,0B6H,0B6H
DB 0B6H,0B6H,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
DB 0,0,0,92H,92H,92H,92H,92H,92H,0B6H,0B6H,0B6H
DB 0B6H,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
DB 0,0,0,0,0,49H,49H,49H,49H,49H,49H
DB 49H,49H,49H,49H,49H,0DBH,0DBH,0FFH
DB 0,0,0,0,0,24H,24H,24H,24H,24H,24H,0B6H
    
```

```

DB 0B6H,0B6H,0B6H,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
YSPU DW OFFSET YSHD2
ZD05H: TEST CS:KPFLAG,1 ;正在打印屏幕
JZ ZD01
IRET
ZD01: OR CS:KPFLAG,1 ;置标志
STI
PUSH DS ;保存中断前的寄存器
PUSH ES
PUSH AX
PUSH BX
PUSH CX
PUSH SI
PUSH DI
PUSH DX
PUSH BP
MOV SI,0
MOV DS,SI
MOV SI,449H
MOV AL,[SI] ;取屏幕方式
CMP AL,13H
JNZ ZD03 ;不是方式13H转
MOV SI,CS
MOV DS,SI
MOV ES,SI
CALL PRTSC13 ;方式13H下拷贝屏幕
ZD02: POP BP ;恢复中断前的寄存器
POP DX
POP DI
POP SI
POP CX
POP BX
POP AX
POP ES
POP DS
AND CS:KPFLAG,NOT 1 ;清打印屏幕标志
IRET ;中断返回
ZD03: POP BP ;恢复中断前各寄存器内容
POP DX
POP DI
POP SI
POP CX
POP BX
POP AX
POP ES
POP DS
AND CS:KPFLAG,NOT 1
JMP CS:ROM05 ;不是方式13H,转原来中断05H
HUANHANG:
MOV AX,1BH
CALL PRINT
MOV AL,"3"
CALL PRINT
MOV AL,24
CALL PRINT
    
```

```

MOV AL,0AH
CALL PRINT
RET
STCOLOR: PUSH AX          ;设置打印机颜色
MOV AL,1BH
CALL PRINT
MOV AL,"r"
CALL PRINT
POP AX
CALL PRINT
RET
TUXY:      ;设置打印机图形方式
PUSH CX
MOV AL,1BH
CALL PRINT
MOV AL,"*"
CALL PRINT
MOV AL,39
CALL PRINT
MOV AL,CL
CALL PRINT
MOV AL,CH
CALL PRINT
POP CX
RET
PRINT: PUSH AX          ;向打印机发送信息
PUSH DX
MOV DX,0
XOR AH,AH
PUSHF
CALL CS:ROM17
POP DX
POP AX
RET
ZHHD: PUSH BX          ;查灰度矩阵转化打印数据
PUSH DI
XOR AH,AH
SHL AX,1
SHL AX,1
ADD DI,AX
MOV BL,CS:YSB1[DI]
XOR BH,BH
ADD BX,CS:YSPU
MOV AL,CS:[BX]
POP DI
POP BX
RET
PRTSC13: ;方式13拷屏
MOV AL,0DH
CALL PRINT ;回车
MOV AL,0AH
CALL PRINT ;换行
MOV DX,OFFSET YSB
MOV BX,0
MOV CX,256
MOV AX,1017H
INT 10H ;读DAC表
MOV SI,OFFSET YSB
MOV CX,256
MOV DI,OFFSET YSB1
PT1301: LODSB ;转换成减色法色度值
NOT AL
AND AX,3FH
SHR AX,1
SHR AX,1
MOV DL,AL
LODSB
NOT AL
AND AX,3FH
SHR AX,1
SHR AX,1
MOV DH,AL
LODSB
NOT AL
AND AX,3FH
SHR AX,1
SHR AX,1
MOV BH,AL
CMP BH,DL
JB PT1302
MOV BH,DL
PT1302: CMP BH,DH
JB PT1303
MOV BH,DH
PT1303: CMP BH,AL
JNZ PT1304
CMP BH,DH
JNZ PT1304
CMP BH,DL
JNZ PT1304
JMP PT1305
PT1304: MOV BH,0
PT1305: SUB AL,BH
STOSB
MOV AL,DH
SUB AL,BH
STOSB
MOV AL,DL
SUB AL,BH
STOSB
MOV AL,BH
STOSB
LOOP PT1301
MOV AX,0A000H ;屏幕显示缓冲区段地址
MOV DS,AX
MOV SI,0
MOV CX,40 ;打印40次
DY001: CALL PRIN13 ;方式13H打印一行
MOV AX,320 * 5 ;调整屏幕缓冲区地址指针
    
```

```

ADD SI,AX
CALL HUANHANG ;换行
LOOP DY001 ;循环
MOV AL,1BH ;以下设置打印机换行距离为30点
CALL PRINT ;30点为标准行距
MOV AL,"3"
CALL PRINT
MOV AL,30
CALL PRINT
MOV AL,0DH ;回车
CALL PRINT
MOV AL,0AH ;换行
CALL PRINT
RET
PRINTH13: ;方式13打印一行
MOV DI,0
MOV AL,4 ;黄色
CALL PT13DH ;打印一种颜色的一行
MOV DI,1
MOV AL,1 ;品红色
CALL PT13DH ;打印一种颜色的一行
MOV DI,2
MOV AL,2 ;天蓝色
CALL PT13DH ;打印一种颜色的一行
MOV DI,3
MOV AL,0 ;黑色
CALL PT13DH ;打印一种颜色的一行
RET
PT13DH: ;方式13打印一种颜色的一行
CALL STCOLOR ;设置颜色,AL中有颜色号
PUSH SI
PUSH DI
PUSH CX
MOV CX,320 * 4
CALL TUXY ;设置图形方式
MOV CX,320
PH131: CALL FSDPD ;发送单排点
INC SI
LOOP PH131
MOV AL,0DH ;回车
CALL PRINT
POP CX
POP DI
POP SI
RET
FSDPD: MOV BH,0 ;发送单排点
MOV CS:YSPU,OFFSET YSHD1
DAYN131:PUSH SI
MOV AL,[SI]
MOV DX,0
MOV BL,0
CALL ZHHD
AND AL,0F8H
OR DH,AL
ADD SI,320
MOV AL,[SI]
CALL ZHHD
PUSH AX
AND AL,7
OR DH,AL
POP AX
AND AL,0C0H
OR DL,AL
ADD SI,320
MOV AL,[SI]
CALL ZHHD
AND AL,3EH
OR DL,AL
ADD SI,320
MOV AL,[SI]
CALL ZHHD
PUSH AX
AND AL,1
OR DL,AL
POP AX
AND AL,0F0H
OR BL,AL
ADD SI,320
MOV AL,[SI]
CALL ZHHD
AND AL,0FH
OR BL,AL
MOV AL,DH
CALL PRINT
MOV AL,DL
CALL PRINT
MOV AL,BL
CALL PRINT
POP SI
ADD CS:YSPU,16
INC BH
CMP BH,4
JZ DY132
JMP DAYN131
DY132: RET
ZK: MOV AX,3505H ;得到原来中断05H入口
INT 21H
MOV WORD PTR ROM05,BX ;保存原来中断05H入口
MOV WORD PTR ROM05[2],ES
MOV AX,2505H ;重新设置中断05H向量
MOV DX,OFFSET ZD05H
INT 21H
MOV DX,OFFSET ZK+1
INT 27H ;程序驻留内存,退出
CODE ENDS
END START

```