

计算机图形与图像丛书

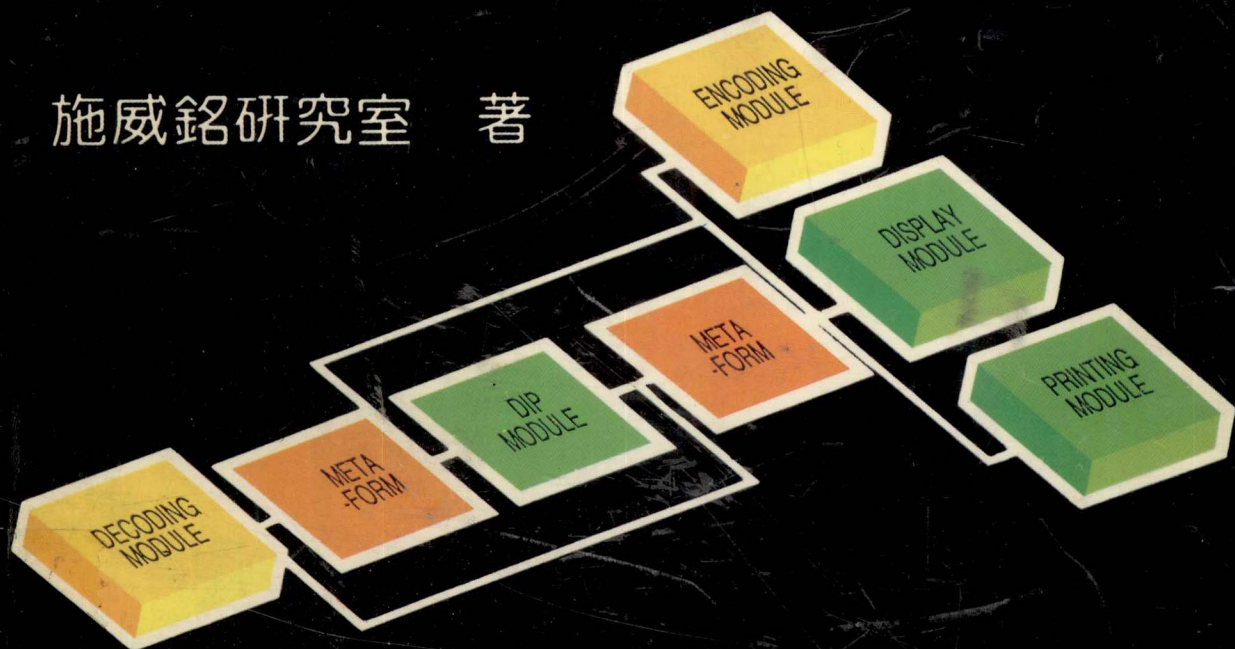


SVGA 显示原理和绘图技巧

SVGA

原理、秀圖、秀字、印字

施威銘研究室 著



旗標



学苑出版社

计算机图形与图像丛书

SVGA 显示原理和绘图技巧

施威铭研究室	编著
方 梅	改编
熊可宜	审校

学苑出版社

(京)新登字 151 号

内 容 提 要

本书详细介绍了 Super VGA 的显示原理和绘图应用。第一篇介绍了 Super VGA 的基本显示原理、VGA/SVGA 的显示模式、视频内存、内存地址与屏幕的对应关系、颜色平面的读写方式、颜色的控制、字型显示的原理与应用等。第二篇主要介绍了编写绘图程序以及显示图形、显示字符、打印图形的技术。另外书中提供了许多有实用价值的程序可供借鉴使用。

需要本书的用户,请直接与北京海淀 8721 信箱书刊部联系,邮政编码:100080,电话:2562329。

版 权 声 明

本书繁体字中文版名为《PC 影像处理技术(三) SVGA 原理、秀图、秀字、印字》,由旗标出版有限公司出版,版权归旗标出版有限公司所有。本书简体字中文版由旗标出版有限公司授权出版。未经出版者书面许可,本书的任何部分不得以任何形式或手段复制或传播。

计算机图形与图像丛书

SVGA 显示原理和绘图技巧

编 著:施威铭研究室

改 编:方 梅

审 校:熊可宜

责任编辑:甄国宪

出版发行:学苑出版社 邮政编码:100036

社 址:北京市海淀区万寿路西街 11 号

印 刷:列电印刷厂

开 本:787×1092 1/16

印 张:25.5 字 数:588 千字

印 数:1~5000 册

版 次:1995 年 11 月北京第 1 版第 1 次

ISBN7-5077-0884-5/TP·26

本册定价:30.00 元

学苑版图书印、装错误可随时退换

前 言

在 PC 这个领域中,学习计算机绘图是一件很辛苦的事。首先遇到的第一个难题是:没有公认的标准。就以视频卡为例,VGA 320×200 256 色的规格根本无法满足大多数绘图软件的需求,而 8514/A 的高价格又不是一般人所能接受的,于是各种低价格的视频卡便几乎席卷了整个市场,这些卡的功能大多数超过了 IBM 制定的规格(VGA,8514/A)。在没有标准可遵循的情况下,芯片设计厂商只有自定规格,也因此造成了程序设计人员无所适从的情况。

第二个难题是:数据取得不易。一个计算机绘图系统至少要包含阅读、显示、打印等功能,要取得各种图像文件规格的数据、各种显示控制芯片的数据以及各式打印机的数据本来就不容易,何况大多数原厂数据都以技术性数据为主,一般用户阅读起来很不容易,往往花了大量的时间阅读也不一定能从中获得想要的信息。

在我们自己的学习过程中,当然也碰到过这些问题。于是,在决定要规划一系列计算机绘图丛书之后,我们花了相当长的时间思考如何建造出一个兼具教学性与实用性的计算机绘图系统。经过一年多的时间,我们创造了 FLIPS。

FLIPS 是一个功能强大的绘图开发环境,它不但具有阅读、显示、打印三大功能,还包含了许多图像处理工具。值得一提的是,我们采用模块化的方式来建造整个 FLIPS 系统,所以,不但所有工具都可以像积木般地自由组合使用,你还可以将自行创造的工具加入 FLIPS 系统。

本书中,我们将介绍 Super VGA 的显示原理以及显示图形、显示字符、打印图形的各种技巧。书中还提供了相当多的程序,有些小程序只是让读者用来做实验,有些则真正具有实用价值。我们希望读者在阅读本书时,能够把每一节程序都拿来玩玩看看,相信必能对书中的内容有更深入的了解。另外,VESA BIOS EXTENSION(简称 VBE)已逐渐成为控制 Super VGA 的标准接口,这在本书中也会详细介绍。

本书在显示图形和打印图形的程序中是采用 FLIPS 的 MetaForm 图像文件格式来运行,所以读者必须对这个格式有所了解才行(在第一章中会介绍 FLIPS 的结构)。当然,我们还是希望你能以“循序渐进”的方式来学习数字图像处理的各个领域。

如果你对 Super VGA 及数字图像处理有兴趣,那么,欢迎你和我们一起投入 FLIPS 的世界。

目 录

第一章 有关计算机图形与图像丛书.....	1
1.1 计算机图形与图像	1
1.2 FLIPS 的整体结构	2
1.3 FLIPS 中各工具的使用方法	8
1.4 MetaForm 的格式	10
1.5 FLIPS. H 包含文件	19

第一篇 显示原理

第二章 基本显示原理	22
2.1 显示卡.....	22
2.2 显示器.....	23
2.3 显示卡的显示原理.....	24
2.4 VGA 卡的内部运行方式	25
2.5 VGA 卡上视频内存的分区	26
2.6 色彩原理.....	26
2.7 VGA 卡上的 RAM DAC	27
2.8 SVGA 卡的加强功能	28
2.9 屏幕坐标.....	30
第三章 VGA/SVGA 的显示模式	32
3.1 VGA 的文本模式	32
3.2 显示页.....	35
3.3 文本模式下的光标.....	37
3.4 VGA 的绘图模式	42
3.5 SVGA 的显示模式	44
第四章 视频内存、内存地址与屏幕的对应关系.....	49
4.1 文本模式下视频内存地址与屏幕的对应关系.....	49
4.2 文本模式下颜色平面与视频内存地址的对应关系.....	53
4.3 文本模式下视频内存地址的显示页划分.....	55
4.4 绘图模式下视频内存地址与屏幕的对应关系.....	57
4.5 交错式对应(模式 4,5 和 6)	58
4.6 平面式对应(除了模式 4,5,6 以外的单色或 16 色模式)	60
4.7 线性对应(模式 13)	63
4.8 SVGA 的区域切换功能	65

4.9	HiColor 及 TrueColor 模式	68
4.10	禁止更改视频内存的内容	72
4.11	禁止屏幕的显示功能	73
第五章	颜色平面的读写方式	75
5.1	颜色平面的 4 个锁存器	75
5.2	与颜色平面读写有关的寄存器	76
5.3	写入模式 0	82
5.4	写入模式 1	86
5.5	写入模式 2	87
5.6	写入模式 3	91
5.7	读取模式 0	91
5.8	读取模式 1	93
5.9	读写模式的基本函数	95
第六章	颜色的控制	98
6.1	调色板寄存器与 DAC 寄存器	98
6.2	存取指定的调色板寄存器	100
6.3	屏幕的边界颜色	102
6.4	一次存取 16 个调色板寄存器及屏幕边界颜色	103
6.5	存取 DAC 寄存器的值	105
6.6	灰度累加	109
6.7	灰度模式	110
第七章	字型显示的原理与应用	112
7.1	BIOS 内建的字符集	112
7.2	读取字型的信息	113
7.3	字符集的存储格式	118
7.4	文字模式下字符集的安装与显示	119
7.5	绘图模式下字符集的安装与显示	122
7.6	BIOS 的字型设置接口总览	122
7.7	文字模式下的字型设置	123
7.8	绘图模式下的字型设置	128
7.9	在文字模式下直接存取颜色平面 2 的字型	129
第八章	SVGA 的统一接口: VESA BIOS Extension	132
8.1	VESA BIOS Extension 概论	132
8.2	存储窗	134
8.3	检查 VBE 调用是否成功	136
8.4	读取 SVGA 卡的信息	137
8.5	读取各显示模式的相关信息	138
8.6	设置显示模式	149
8.7	读取当前的显示模式	150

8.8 存储或恢复 SVGA 的视频状态	152
8.9 存储窗的区域切换控制	155
8.10 屏幕垂直、水平快速滚动的功能	156
8.11 设置或读取 DAC 各原色的有效位数	161

第二篇 绘图应用

第九章 编写绘图程序的技巧	164
9.1 加快程序速度的技巧	164
9.2 使用汇编语言	167
9.3 C 与汇编语言间的参数传递	167
9.4 在线汇编语言	172
9.5 在绘图模式下的检错	174
第十章 建立绘图环境	176
10.1 绘图函数库的目标与结构	176
10.2 编译器设置	177
10.3 绘图函数库的主要包含文件	177
10.4 绘图函数库的程序结构	183
10.5 GrVga.h 和 GrVga.c	186
10.6 GrVesa.h 和 GrVesa.c	195
10.7 GrUtl.c	206
10.8 GrLib.c	210
第十一章 显示图形	227
11.1 绘图工具——GrDraw.c	227
11.2 显示图形模块——GrShow.c	237
11.3 键盘输入	243
11.4 测试绘图函数库的程序——GrTest.c	246
11.5 显示图形主程序——M2Svga.c	251
第十二章 显示字符	255
12.1 BIOS 提供的显示字符功能	255
12.2 存储字体信息的结构体	258
12.3 两个通用的函数	259
12.4 读取 BIOS 内建的字体	260
12.5 由英文字体文件读取字体	261
12.6 读取中文字体的原理	262
12.7 如何读取倚天中文字体	266
12.8 关闭已打开的字体	269
12.9 显示字体	270
12.10 基本的字体变化	270

12.11	进阶的字体变化	275
12.12	显示中、英文字符串	279
12.13	完整的字符显示模块	280
12.14	测试程序	295
12.15	其他的字体文件	300
12.16	制作自己的点阵字体文件	301
第十三章	打印图形	307
13.1	打印机概论	307
13.2	ESC/P 及 PCL 语言简介	310
13.3	PostScript 语言简介	310
13.4	颜色处理	312
13.5	点阵打印机的打印原理	315
13.6	喷墨打印机的打印原理	317
13.7	激光打印机的打印原理	319
13.8	与打印机交互	319
13.9	使用 ESC/P 语言来打印图形	324
13.10	ESC/P 的打印图形程序	328
13.11	使用 PCL 语言来打印图形	338
13.12	PCL 的打印图形程序	339
13.13	ESC/P2 新增的打印图形功能	345
附录 A	本书配套磁盘的内容	347
附录 B	BIOS INT 10h 常用功能一览表	354
附录 C	BIOS 预设的 4 组 DAC 颜色值	356
附录 D	曾世卡(Tseng Lab)的驱动程序	368
附录 E	Trident 显示卡的驱动程序	386

第一章 有关计算机图形与图像丛书

本书的内容主要是介绍 VGA、SuperVGA 的显示原理以及显示图形、显示字符与打印图形的技巧。虽然就内容来说本书是完全独立的,但也同时是“计算机图形与图像丛书”的一部分。在本章中,我们先介绍一下贯穿本书的概念:FLIPS 结构。

1.1 计算机图形与图像丛书

“计算机图像处理”领域非常广阔,凡是与图像有关的课题都包含在内。不过我们还可以将它大致地分成 5 个部分:

1. 阅读各种图像文件,例如 PCX、GIF、BMP 等。
2. 转换不同格式的图像文件,例如将 PCX 转成 GIF 文件。
3. 将各种图像文件显示在屏幕上。
4. 打印各种图像数据。
5. 数字图像处理,例如调整图形的亮度、对比度、清晰度等。

事实上,前 4 项只能算是图像处理的“前端处理”(front end)工作,而第 5 项“数字图像处理”(DIP, Digital Image Processing)才是真正拿图像来做一些特殊效果的处理。话虽如此,这 5 项却都是同样重要的,而且缺一不可。

在此,旗标公司设计了一个小型的图像处理系统,那就是 FLIPS (Flag Image Processing System),并且根据结构编写出一系列的书来介绍上述 5 个主题,其内容分别如下:

1. PC 图像处理技术(一):图形文件压缩篇(已出版)

探讨数据压缩原理以及 IMG、MAC、GIF、PCX 图像文件。

2. PC 图像处理技术(二):文件压缩续篇(已出版)

探讨数据压缩原理以及 BMP、TGA、TIF、EPS、JPG 图像文件。

上述两本书已由学苑出版社合并改编出版为《PC 图像处理技巧与实例》。需要该书的读者,可直接与北京海淀 8721 信箱书刊部联系,邮编:100080,电话:2562329,传真:2579874。

3. PC 图像处理技术(三):SVGA 原理、秀图、秀字、印字

探讨 Super VGA 的显示原理、显示图形、显示字符与打印图形(改编成本书)。

4. PC 图像处理技术(四):(规划中)

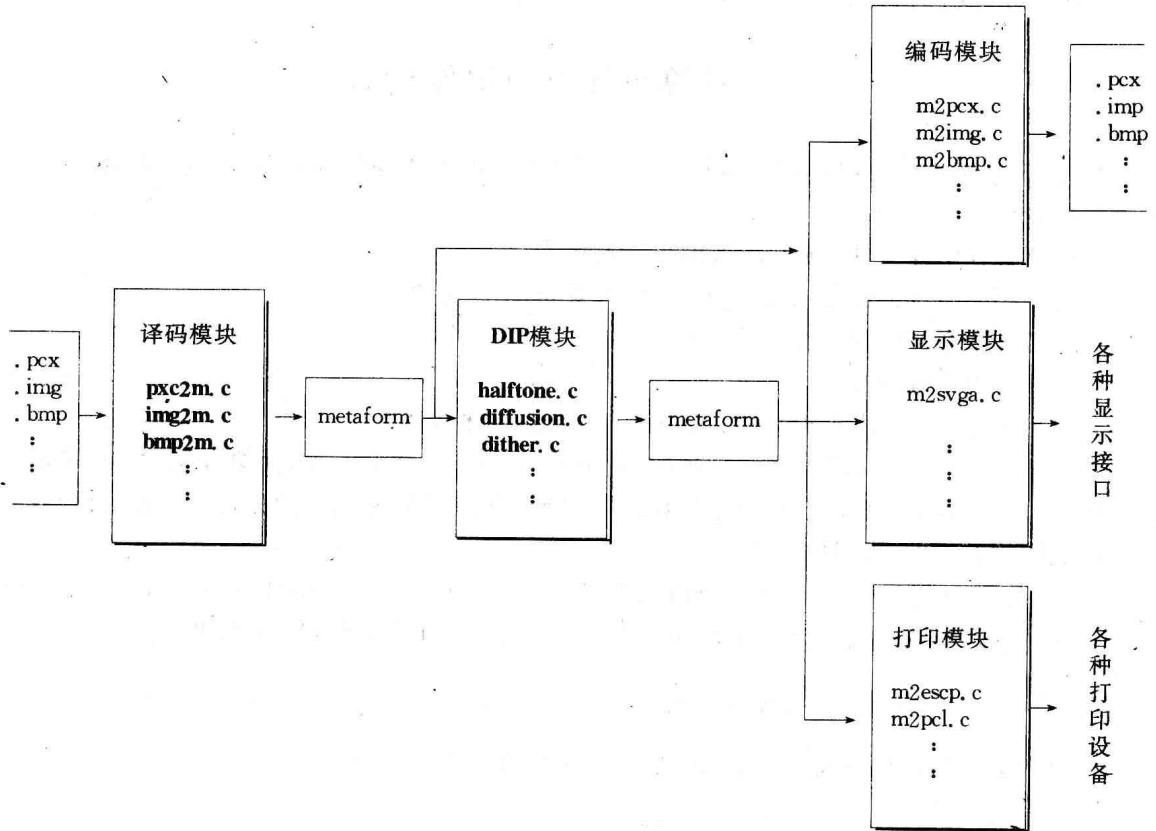
探讨各种数字图像处理的技术与应用。

当然,购买本书并不表示你也必须购买本系列其他的书。如果读者只对本书的内容有兴趣,那么只阅读本书就可以了,不过本书在显示图形和打印图形的程序中是采用 FLIPS 的

MetaForm 图形文件格式来运行,所以读者必须对这个格式有所了解才行(后面会详细介绍)。

在《PC 图像处理技巧与实例》一书中已对 FLIPS 及 MetaForm 做了详细介绍,不过其中有一小部分未能切合实际需求,所以在本书中已做了一些修改(HiColor 及 TrueColor 部分)。接着,就让我们来看看 FLIPS 的结构和 MetaForm 的格式吧!

1.2 FLIPS 的整体结构



上图是 FLIPS 的方块图,由图中我们可以看出 FLIPS 一共具备了 5 个模块(Modules):

- ☐ 译码模块:在《PC 图像处理技巧与实例》一书中介绍
- ☐ 编码模块:在《PC 图像处理技巧与实例》一书中介绍
- ☐ 显示模块:在本书中介绍
- ☐ 打印模块:在本书中介绍
- ☐ DIP(数字图像处理模块)

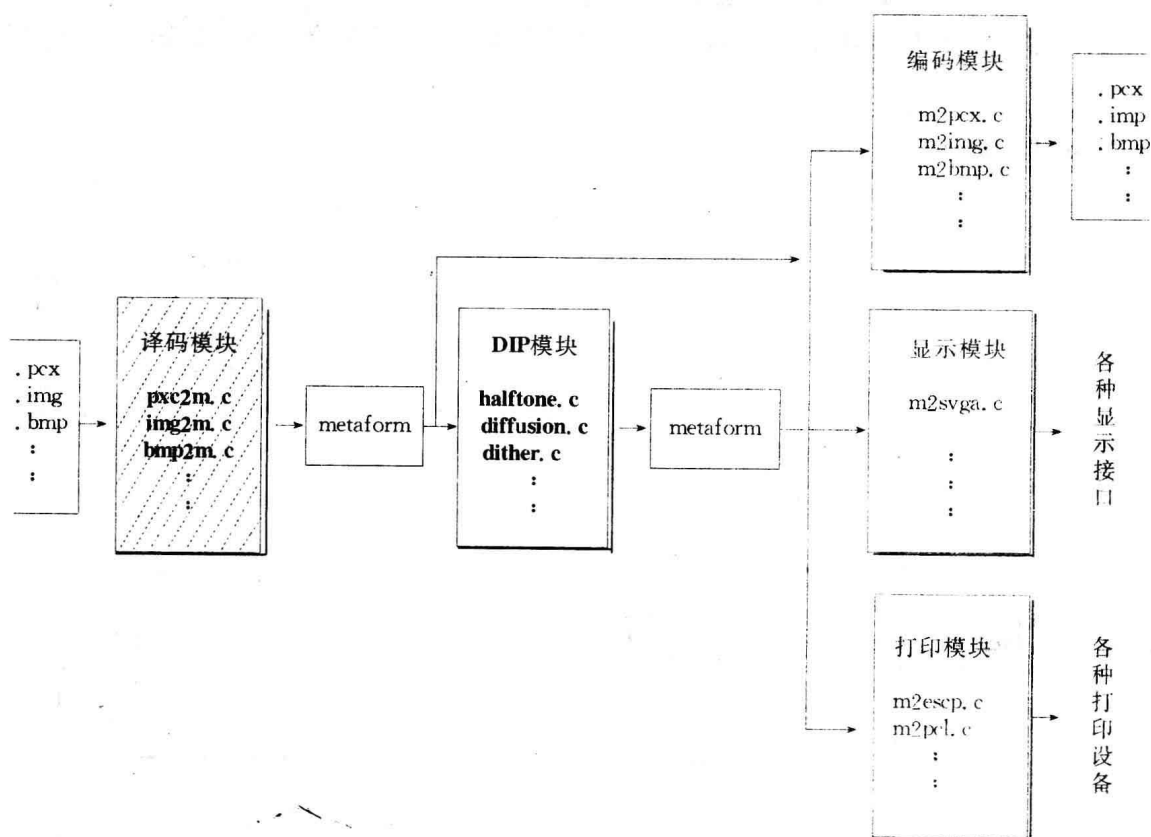
还有一个通用图像格式 MetaForm。我们在后面会介绍各模块及 MetaForm 的功能。

FLIPS 的工作原理基本上来说相当简单,所有的程序都以 MetaForm 为界限,属于译码模块的程序就只需要负责将图像文件转换成 MetaForm,而属于显示模块的程序就只需要负责显示 MetaForm 的图形就可以了。因此,每一个程序的功能都相当单纯且容易设计,但是组合起来却能发挥强大的功能。

2. 易处理性: MetaForm 是一种非压缩格式, 每一个像素至少占一个字节, 所以在做各种像素运算时非常方便。但是你应该注意到, 任何图像文件转换成 MetaForm 后, 几乎都会占用比原来还要大的空间。不过对于 FLIPS 来说这不是问题, MetaForm 是一种“数据交换格式”(Data Exchange Meta Format)而非“数据存储格式”(Data Storage Format), 它只是各种处理过程的中间产物, 而不是最后输出结果, 所以存储效率对 MetaForm 来说并不是很重要。

我们将在 1.4 节中详细介绍 MetaForm 的数据格式。

译码模块



译码模块负责将各种不同格式的图像转换为 MetaForm。由于这个转换过程相当于经过压缩的各种图像文件转换为没有经过压缩的 MetaForm, 所以你也可以将译码模块称之为解压缩模块。

由于每一个图形文件格式会有一个转换程序, 所以译码模块中包含了许多程序, 这些程序全部根据下列的原则来命名:

xxx2m.c

其中 xxx 代表要解码的文件格式, m 代表转换成 MetaForm, 而“2”当然就是代表“to(到)”了, 所以

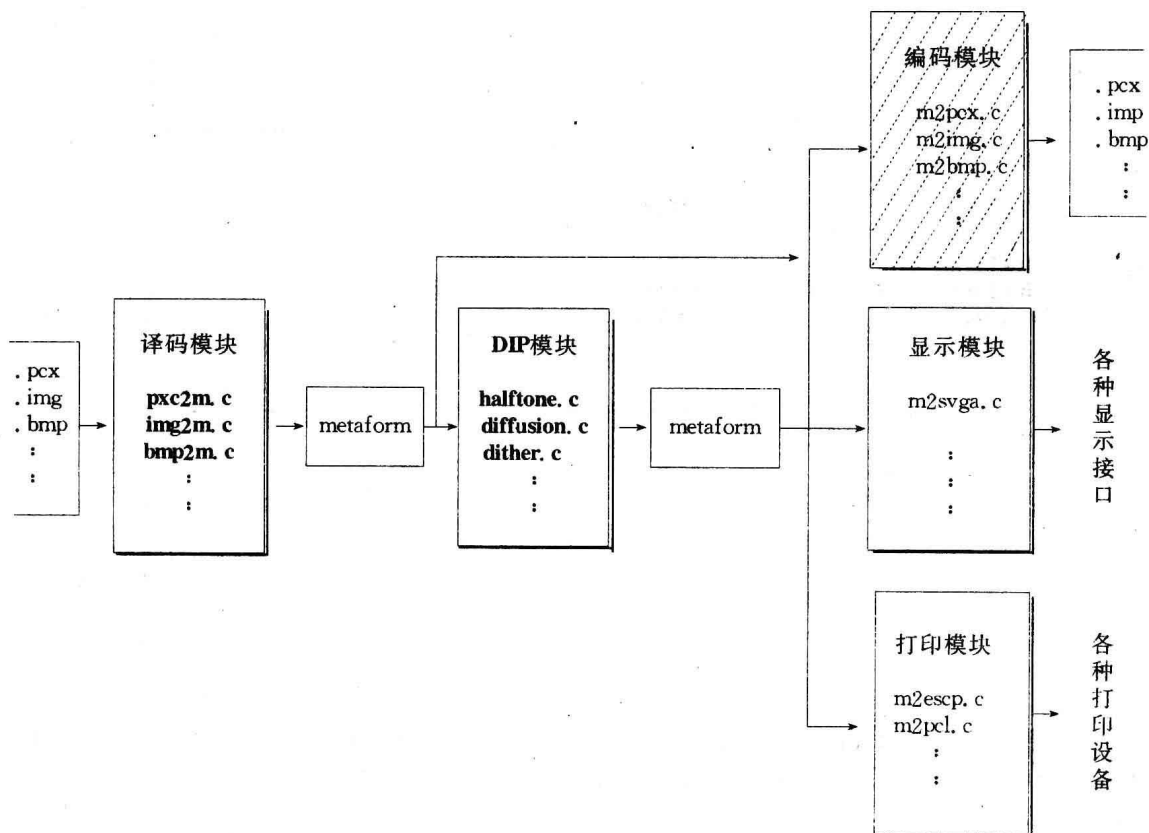
mac2m.c

就是代表将 MacPaint 格式的图像转换为 MetaForm 的程序。同理可知：

pcx2m. c

就是代表将 PCX 格式的图像转换为 MetaForm 的程序。

编码模块



编码模块的功能和译码模块正好相反,它负责将 MetaForm 转换为各种图像格式文件。通常在经过此转换后,文件尺寸会明显变小(因为压缩的原因),所以编码模块也可称之为压缩模块。

Encoding 模块中的程序命名方式和译码模块类似:

m2xxx. c

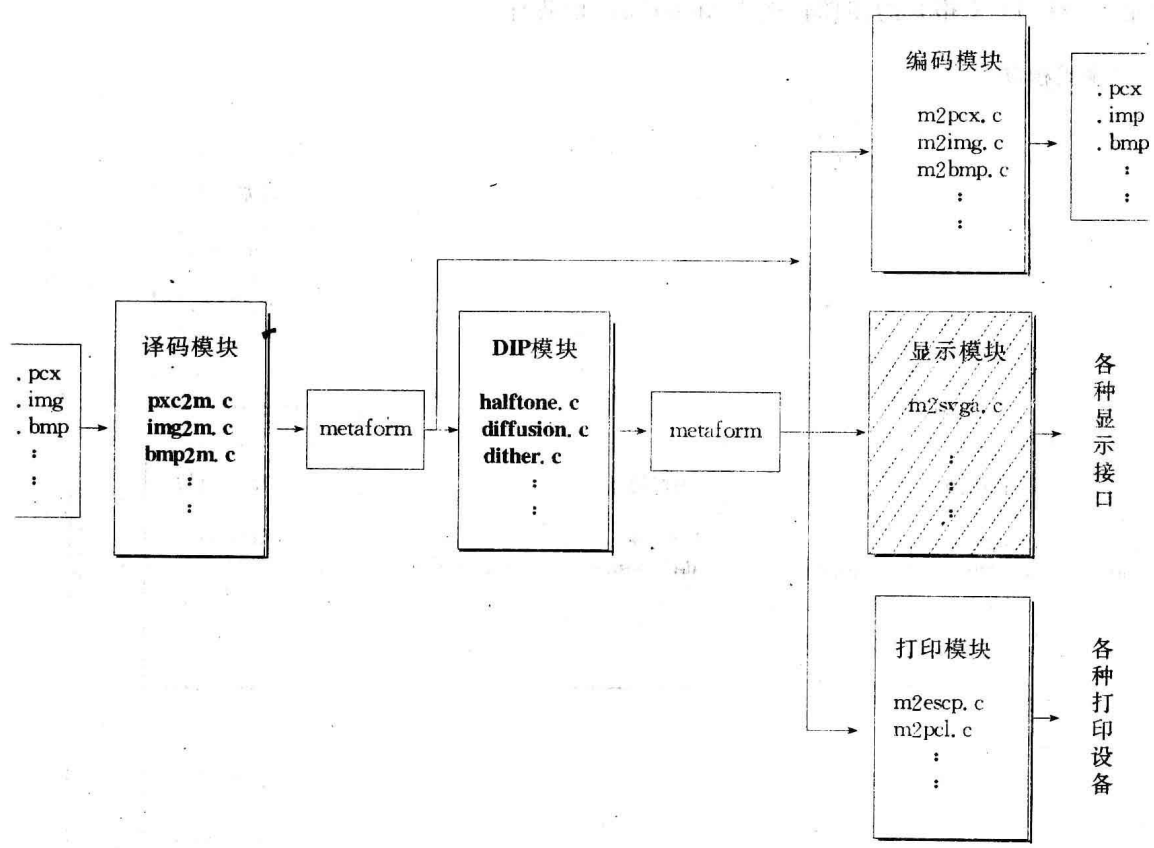
其中 m 代表 MetaForm,而 xxx 则代表要编码成的文件格式,所以:

m2pcx. c

就是代表将 MetaForm 转换为 PCX 文件的程序。

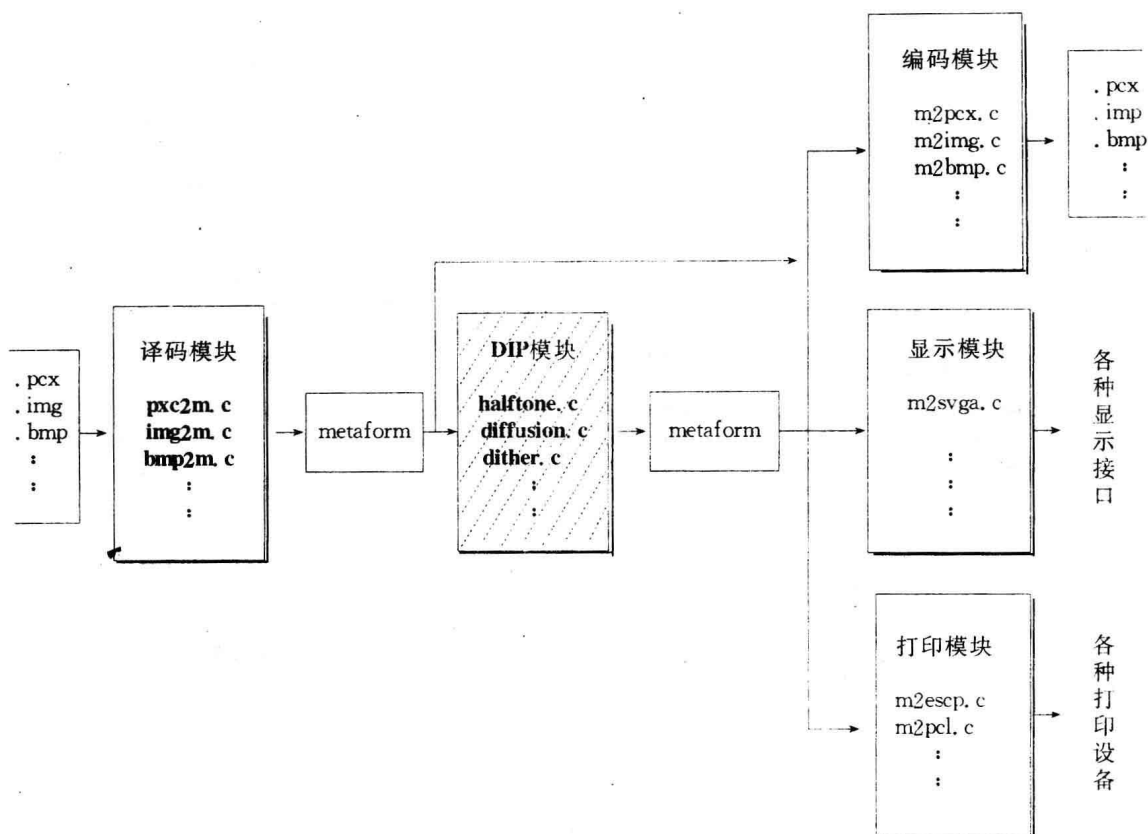
以上我们所介绍的译码模块以及编码模块基本上都是与设备无关的,也就是说你可以轻易地将属于这两个模块的程序移植到 DOS 以外的操作系统上而不需要做重大的修改,但是接下来我们所要介绍的显示模块和打印模块就和你所使用的设备有绝对的关联性了。

显示模块



显示模块负责将 MetaForm 显示到显示设备上。在本书中提供了一个 m2svga.exe 程序，可以接受 MetaForm 格式的数据，然后在 VGA 或 SVGA 的屏幕上显示图形。

DIP 模块



数字图像处理模块(简称 DIP 模块)是整个 FLIPS 的核心。我们在前面介绍了多个模块的目的就是为了要建立一个 DIP 的框架,而 DIP 模块就是这个框架的真正用户。

DIP 模块中包含了很多处理数字图像的工具。例如当我们用扫描仪扫描一张照片并保存成任何一种 FLIPS 可以接受的格式后,接下来我们就可以利用 DIP 模块中的工具来调整这张照片的亮度(Intensity)、对比度(Contrast)以及 γ 值(特性曲线值)等。DIP 模块中还包含了消除图像中噪声(noise)的工具、产生柔焦效果的工具、使图像变得更加清晰(sharp)的工具以及各种用来产生特殊效果(例如:马赛克效果以及线条化效果)的过滤器。此外 DIP 模块还提供了平滑放大、平滑缩小的工具,以便用来调整图像的大小。

1.3 FLIPS 中各工具的使用方法

基本上,FLIPS 中的所有工具都遵循以下的设计原则:程序中所有的 I/O 操作都通过标准 I/O(Standard Input/Output)来进行。遵循此原则设计的程序称之为过滤器(filter),所以我们可以说 FLIPS 是由一大堆的过滤器所构成。你是否会好奇为什么我们要这样设计程序呢?在回答这个问题前让我们先介绍一些重要的概念:I/O 转向(I/O redirection)和管道(Pipe)。

DOS 将键盘与屏幕称之为标准 I/O,凡是通过标准 I/O 进行输入输出的程序都可用转向

的方式来改变成由其他文件做输入输出。DOS 的 I/O 转向是通过“>”以及“<”来完成的。例如,执行 dir 可在屏幕上显示当前目录的内容,而我們也可以通过 I/O 转向的方式将 dir 的结果转到另一个文件:

```
C:\>dir > test.dir
```

那么 dir 的所有输出都将转存到 test.dir 中,而不会显示到屏幕上了。

至于 pipe,则是通过“|”将某一个程序的输出当作另一个程序的输入,例如:

```
C:\>dir | more
```

就会将 dir 的输出结果转向给 more(作为 more 的输入),所以你就可以一页一页来看 dir 的结果了。

我们之所以要将所有程序写成 filter 的类型,主要就是为了要达到管道及 I/O 重定向的目的。如此一来,模块和模块之间要做数据的交换也就变得相当容易了。接下来就让我们来看几个 FLIPS 的使用实例。

假设当前我们已经完成了下列程序:

- ☐ pcx2m.exe:将 PCX 转换为 MetaForm。
- ☐ m2bmp.exe:将 MetaForm 转换为 BMP。
- ☐ m2svga.exe:将 MetaForm 显示在屏幕上。
- ☐ m2pcl.exe:将单色的 MetaForm 在 PCL 兼容的打印机上打印出来。
- ☐ dither.exe:将彩色 MetaForm 转换为单色 MetaForm。

那么当我们执行:

```
C:\>pcx2m < test.pcx | m2svga
```

首先 pcx2m 会通过“<”读入 test.pcx,然后将它转换成 MetaForm,接着再通过“|”将这个 MetaForm 传给 m2svga 作为输入数据,于是最后 test.pcx 就会被显示在屏幕上了。事实上,我们将上述命令拆开来执行:

```
C:\>pcx2m < test.pcx >tmp.m
```

```
C:\>m2svga <tmp.m
```

也可以得到相同的结果(不过多了要删除 tmp.m 的麻烦)。同样的道理:

```
C:\>pcx2m < test.pcx | m2bmp > test.bmp
```

可以将 PCX 格式的 test.pcx 转换为 BMP 格式的 test.bmp。而:

```
C:\>pcx2m < test.pcx | dither | m2pcl
```

就较为复杂了,首先 pcx2m 读入 test.pcx 并将其转换为 MetaForm,接着涂色过程通过管道读入此 MetaForm,并且将其转换为单色图形的 MetaForm,最后 m2pcl 读入涂色的结果,并将其转换为 HP PCL 格式而送给 PCL 兼容打印机打印出结果。

有关 m2svga 的详细用法,请参考 11.5 节的说明;m2escp 及 m2pcl 的用法则请参考 13.10、13.12 节的说明。