

# EGA/VGA/TVGA 开发指南

刘宏程 张 鹏 等 编著

电子工业出版社

771654

# EGA/VGA/TVGA 开发指南

刘宏程 张 鹏 等编著



电子工业出版社

(京)新登字 055 号

图书在版编目 (CIP) 数据

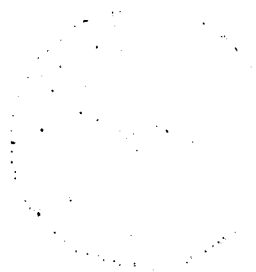
EGA/VGA/TVGA 开发指南 / 刘宏程, 张鹏编著. - 北京:  
电子工业出版社, 1994. 6

ISBN 7-5053-2199-4/TP·566

I. E... II. ①刘... ②张... III. 程序设计方法学 - EGA,  
VGA, TVGA IV. TP311.11

中国版本图书馆 CIP 数据核字 (94) 第 05427 号

ISBN 7-5053-2199-4



电子工业出版社出版

北京市海淀区万寿路 173 信箱 (100036)

电子工业出版社发行 各地新华书店经售

北京天利电子出版技术公司排版

北京市顺义县天竺颖华印刷厂印刷

\*

开本: 787×1092 毫米 1/16 印张: 32.75 字数: 800 千字

1994 年 6 月第 1 版, 1994 年 6 月第 1 次印刷

印数: 0,001-3,500 册 定价: 36.00 元

## 前 言

微型计算机显示技术广泛地应用于微机的各个应用领域。随着 EGA/VGA/TVGA 适配器在全国的普及推广,如何有效地利用系统提供的各种资料,开发出能满足各类实际应用要求的系统及应用软件,是广大计算机科研和应用人员面临的最迫切的问题。

要充分利用 EGA/VGA/TVGA 提供的良好环境来开发好的文本、图形应用软件,就必须深入了解和掌握 EGA/VGA/TVGA 的知识和应用技术。但是,国内这方面的资料缺乏,尤其是有关 TVGA 的资料很少。国外的某些资料,对一般开发人员来讲又难以理解,而且内容较为分散,不易查找所需的技术信息,因此,开发人员很难深入到系统内部。另一方面,多数计述图形学的书籍又与现有技术存在一定的差距,难以与 EGA/VGA/TVGA 结合,而计述 EGA/VGA 的资料也未涉及这类问题,给出的实便较少或效率不高,难以用于实际的软件开发。

本书的目的就是为了要解决上面这两方面的问题,为程序开发人员提供一份较为详尽的 EGA/VGA/TVGA 技术资料,并尽量缩小理论和应用之间的差距。

本书共分三篇,第一篇为原理篇,主要计述图形显技术、图形学基本原理和常用算法,重点是在实际开发中应用较为广泛的图形学算法和图象压缩算法;第二篇为技术篇,该篇系统地分析了 EGA/VGA/TVGA 的基础知识、体系结构、控制寄存器结构和功能以及 BIOS 的结构、功能调用技术,同时还讨论了 EGA、VGA、TVGA 三者之间的兼容性问题;第三篇为应用篇,以大量的汇编语言实例说明 EGA/VGA/TVGA 程序的设计技巧,这些实例包括了在大多数实际应用中的子程序,如画点、线、圆、平滑滚动、屏幕分割、图形图象的压缩和解压缩等子程,这些实例力图充分利用 EGA/VGA/TVGA 的各种资源,来提供效率,缩短图形学算法和实际应用的差距,并有助于加深理解前两篇的一些内容。

我们是长期从事计算机应用软件开发人员,在编写本书过程中,曾对许多概念的理解上多次推敲、测试、力求做到概念清楚,数据可靠,书中所附实例也多是作者在实际应用中一些子程序,均可以直接引用。

本书适合于程序开发人员阅读,对于从事计算机界面设计和图形软件设计的工作人员有很好的参考价值,也可作为大专院校计算机、自动化、无线电、建筑等专业的教学参考书。

另外,本书配有包含书中所有例子程序以及符合 C、PASCAL 调用格式的子程序的磁盘,该软件由电子工业出版社同时出版发行。

参加本书编写的还有江南、吴明和崔军等三位同志。由于时间仓促以及编者的水平有限,书中的错误和不妥之外在所难免,敬请读者不吝批评指教。

编者 1993 年 10 月

# 目 录

## 第一篇 原理篇

### 前言

<b>第一章 图形学基础</b> .....	1
§ 1.1 计算机图形学的发展与应用 .....	1
§ 1.2 点生成算法 .....	6
§ 1.3 二维几何变换 .....	16
§ 1.4 三维几何变换 .....	26
§ 1.5 观察变换 .....	29
§ 1.6 图形的裁剪 .....	35
§ 1.7 色彩模型 .....	37
§ 1.8 图象压缩 .....	39

## 第二篇 技术篇

<b>第二章 EGA/VGA/TVGA 体系结构</b> .....	42
§ 2.1 显示器 .....	42
§ 2.2 EGA/VGA/TVGA 体系结构 .....	46
§ 2.3 文本显示模式 .....	49
§ 2.4 图形显示模式 .....	58
§ 2.5 调色板和 DAC 彩色寄存器 .....	74
<b>第三章 EGA/VGA/TVGA 寄存器</b> .....	78
§ 3.1 EGA/VGA/TVGA 寄存器概述 .....	78
§ 3.2 外部寄存器组 .....	79
§ 3.3 CRT 控制寄存器 .....	83
§ 3.4 图形控制寄存器 .....	94
§ 3.5 操作定序寄存器 .....	99
§ 3.6 属性控制器寄存器 .....	102
§ 3.7 DAC 彩色寄存器 .....	106
§ 3.8 TVGA 扩充寄存器 .....	108
§ 3.9 将 TVGA 设置成 EGA、CGA 和 MDA 的方法 .....	115
<b>第四章 EGA/VGA/TVGA BIOS</b> .....	117
§ 4.1 EGA/VGA/TVGA BIOS 描述 .....	117
§ 4.2 BIOS 数据区 .....	117

§ 4.3	<b>EGA/VGA/TVGA 环境表</b>	119
§ 4.4	<b>BIOS 基本功能表</b>	122
§ 4.5	<b>调色板及 DAC 处理功能</b>	133
§ 4.6	<b>字符集处理功能</b>	141
§ 4.6	<b>BIOS 其它功能</b>	151

## 第三篇 应用篇

<b>第五章 绪论</b>	167
§ 5.1 阅读说明	167
§ 5.2 磁盘组织	170
<b>第六章 基本操作</b>	172
§ 6.1 写寄存器	172
§ 6.2 读寄存器	175
§ 6.3 取显示模式	176
§ 6.4 取文本分辨率	177
§ 6.5 取光标类型	178
§ 6.6 取当前活动显示页	180
§ 6.7 取光标位置	181
§ 6.8 取显示页长度	185
§ 6.9 取显示适配器类型	186
§ 6.10 取显示器类型	189
§ 6.11 取显示存储器容量	191
§ 6.12 取字符高度	193
§ 6.13 取字符宽度	194
§ 6.14 取扫描线数目	195
§ 6.15 选择 <b>TVGA</b> 页模式	196
§ 6.16 设置 <b>TVGA</b> 的 <b>BPS</b>	197
§ 6.17 读扫描方式	199
§ 6.18 读位平面数	200
<b>第七章 文本操作</b>	202
§ 7.1 设置光标位置	202
§ 7.2 设置光标类型	204
§ 7.3 滚动文本窗口	206
§ 7.4 写字符	211
§ 7.5 读字符	217
§ 7.6 允许与禁止文本闪烁	218
§ 7.7 显示页切换	222
§ 7.8 16 和 32 个显示页	224
§ 7.9 文本屏幕分割	233
§ 7.10 设置逻辑屏	237

§ 7.11	文本模式下的平滑滚屏 .....	237
<b>第八章</b>	<b>字符发生器编程</b> .....	<b>250</b>
§ 8.1	读字符发生器 .....	250
§ 8.2	写字符发生器 .....	254
§ 8.3	512 字符集 .....	260
§ 8.4	43/50 行文本显示 .....	265
§ 8.5	用户自定义字符集 .....	266
<b>第九章</b>	<b>图形模式基本操作</b> .....	<b>273</b>
§ 9.1	初始化图形模式 .....	273
§ 9.2	写象素 .....	274
§ 9.3	读象素 .....	285
§ 9.4	画线 .....	287
§ 9.5	造型线 .....	304
§ 9.6	矩形 .....	319
§ 9.7	画圆 .....	320
§ 9.8	画椭圆 .....	326
§ 9.9	造型扫描线 .....	336
§ 9.10	矩形填充 .....	343
§ 9.11	圆填充 .....	351
§ 9.12	椭圆填充 .....	354
§ 9.13	显示字符 .....	359
§ 9.14	块传送 .....	365
§ 9.15	图形转贮 .....	385
§ 9.16	图形显示 .....	392
§ 9.17	显示页切换 .....	404
§ 9.18	读 DAC 彩色寄存器 .....	405
§ 9.19	写 DAC 彩色寄存器 .....	407
<b>第十章</b>	<b>高级图形操作</b> .....	<b>409</b>
§ 10.1	设置逻辑屏 .....	409
§ 10.2	平滑滚屏 .....	410
§ 10.3	屏幕分割 .....	426
§ 10.4	设置非标准显示模式 .....	428
§ 10.5	<b>HSV 彩色与 RGB 彩色的相互转换</b> .....	<b>440</b>
§ 10.6	图象压缩—— <b>GIF 解码</b> .....	<b>443</b>
§ 10.7	图象压缩—— <b>GIF 编码</b> .....	<b>474</b>
<b>【附录】</b>		
<b>A.</b>	<b>寄存器地址、功能索引表</b> .....	<b>491</b>
<b>B.</b>	<b>EGA/VGA/TVGA BIOS 功能一览表</b> .....	<b>494</b>
<b>C.</b>	<b>EGA/VGA/TVGA 标准显示模式</b> .....	<b>496</b>
<b>D.</b>	<b>EGA/VGA/TVGA 寄存器缺省值</b> .....	<b>498</b>

<b>E. EGA/VGA/TVGA 各显示模式的同步信号和极性</b> .....	506
<b>F. 典型显示存储器射图</b> .....	508



# 第一篇 原理篇

## 第一章 图形学基础

随着计算机技术的发展,微机图形显示功能日益提高,现在利用微机已经可以实现以前只有在大型和小型计算机上才能出现的高质量彩色图形。由于计算机图形显示具有直观、方便的特点,所以与计算机图形显示有关的各种应用日益增多。

本章简要地介绍图形学中的基本知识,它将有利于开发出高质量的图形应用软件。

### § 1.1 计算机图形学的发展与应用

计算机图形学是伴随着计算机及其外部设备而产生和发展起来的。

1950年,美国麻省理工学院研制出 Whirlwind I 计算机,它的附件是一个类似示波器阴极射线管的显示器,可以显示一些简单的图形,这就是计算机图形显示的开端。50年代,计算机所配备的图形设备仅仅具有图形输出功能,应用也不广泛;60年代计算机图形学得到蓬勃发展;而70年代,交互式计算机图形学在军事、工业、教育等领域得到了广泛的应用。

目前,计算机图形处理系统种类繁多,其应用已遍及科学技术各个领域。某些部门的设计工作,如飞机、汽车、船舶、大规模集成电路等的设计,离开了 CAD 就寸步难行。

交互式计算机图形学之所以发展迅速,是因为图形显示器与一般的外部设备相比,具有速度快、直观、使用方便等优点。特别是在交互式系统中,使用人员可以利用键盘、鼠标等输入设备,方便地与计算机进行对话,输入数据,修改设计。

#### 一、交互式计算机图形系统的应用领域

由于计算机图形设备的不断更新和图形软件系统的不断完善,使得计算机图形系统的应用领域越来越广泛,而且还在不断扩大。目前,其应用领域包括:

##### 1. 过程监控

各种实时过程,如火箭的运行,某种物理过程,常常可以用一定的曲线来表示过程进行的情况,且由某些参数控制过程的进行。利用交互式计算机图形可以把描述过程的曲线(例如,火箭运行的轨迹)实时显示出来。如果不符合原定要求,可通过反馈(即改变控制参数的值)控制过程的进行,这比用各种仪表来观察和控制过程的进行自然方便、有效得多。

##### 2. 信息资料的检索、处理

在各种存贮大量信息资料的场合(例如图书馆、银行、气象站、各种档案管理室等),利用交互式计算机图形处理系统可把文字或数字信息的管理(通常的数据库系统)与图形处理结合起来,以迅速而形象地获得所需的信息。例如:

把各种地图及有关的地形资料存入计算机,指挥员就可迅速检索到所需的图表资料,并

可看到相应的地形、地貌。

把百科全书存入计算机内,读者可以迅速查阅各种资料,包括其中的图表。

在医疗方面,除可快速查阅各种病案资料外,还可查阅各种 X 射线照片、心电图、脑电图等,并根据有病器官的测量数据形成立体图象,使其旋转、放大等等,以便进行诊断。

由于计算机能快速处理大量信息,因此,还可实现信息资料的动态检索。例如,根据气象资料可以将各种曲线、曲面显示出来,并且表示出气候随时间改变的情况。

### 3. 计算机辅助设计和计算机辅助制造

计算机辅助设计是一个十分广泛的应用领域。现列举以下几个方面:

#### 1) 计算机辅助制图

工程制图、机械制图历来是工程、机械设计中一项耗费大、效率低的工作。因此,实现计算机制图是把人们从繁琐、复杂的劳动中解放出来的有效途径。把各种常用的图形事先存入计算机(图形零件库),就可由设计者随时调用。图形的比例、平移、旋转等等变换,由手工来完成十分繁琐,而计算机只要经过简单的计算就能实现,因此,用计算机控制数控绘图机进行绘图能大大提高工作效率。

但是,没有交互功能的计算机绘图系统仍然是比较初级的系统,因为图形得在程序中事先设计好,而这不是一件轻松的工作,很难做到不出差错。出了差错又常常要在绘图机上绘出之后才能发现,于是必须修改程序,重新做起。如果利用交互式图形系统,设计者就能在荧光屏上设计图形。如不合适,可随时修改,满足要求后再在绘图机上绘出,甚至可在显示屏上直接摄像。这样,自然大大提高了效率。

#### 2) 计算机辅助几何设计

在飞机、汽车、船舶等设计中,外形设计是十分重要的,它既涉及机械、力学等性能要求,又涉及美学方面的要求。全部由手工完成效率低、费时多。但完全由计算机进行自动设计,目前还难以做到。涉及美学方面的问题需由设计人员参与才能解决,因为怎样才美是要由人来判断的。因此,由人和计算机共同进行设计是最为合适的。计算机可以根据输入参数和一定的数学模型进行初步设计,并在屏幕上显示出来。设计者交互地进行修改,再让计算机显示新的外形,再修改,……,直至满意。

#### 3) 计算机辅助结构设计

在机械结构、飞机、汽车、船舶的框架结构,以及建筑、桥梁等的结构设计中,利用交互式图形系统一方面可计算其力学性能,另一方面可显示各种结构的形状与性能曲线,并快速而形象地表示出参数的改变对性能的影响。

#### 4) 计算机辅助线路、管道设计

这方面的例子有印刷线路板布局与布线设计,各种管道的平面或空间的布置设计,交通网络的设计,等等。

#### 5) 计算机辅助电路设计与逻辑设计

根据一定的指标,设计电路的参数,并显示相应的特性曲线。或者,把逻辑图转化为电路图,等等。

#### 6) 建筑物的总体设计

各种设计方案无需制作模型即可显示出外形图、各种轴侧图、透视图等等,甚至可以“走

进”建筑物内部去“看看”采光是否良好。至于建筑物的色彩设计,对彩色图形处理系统来说非常方便。

#### 7) 计算机辅助排料和落料

这方面的例子更多,如服装设计中的开片与排料,飞机、汽车、船舶设计中的外板展开与落料,等等。

应该注意到,在这类问题中,常常可以把设计与加工连贯起来。例如服装开片排好后可以通过数控切割机进行切割,这样就成为计算机辅助设计与制造了。

### 4. 计算机辅助教学

教学需要模型、挂图,而利用交互式图形系统可以产生十分理想的模型与生动活泼的挂图,有如科教电影。而且,它远比科教电影制作方便,实用灵活,因为教师可以根据实际情况选择和变换所需画面。

### 5. 计算机模拟与仿真

计算机模拟与仿真方面的应用也很广。例如:

用计算机显示飞行员在飞行中看到的外景,且景象随着操纵杆动作能够变化。同时,在模拟的仪表上显示出各种飞行数据。这就是飞行员的模拟训练器。

用计算机显示模拟的“靶机”和“导弹”,进行计算机辅助打靶训练。

在机械加工中,可以用计算机模拟刀具和工件,并选择适当的刀具或工件轨迹。这就是计算机模拟机械加工。

在理论研究中,交互式图形系统也有许多应用。例如,高分子结构模型的立体显示、受控热核反应的状态显示,等等。

### 6. 计算机美术创作

把计算机显示技术与人工智能技术结合起来,就可以用计算机进行美术图案、绘画等艺术创作。

### 7. 计算机动画制作

在动画片的制作中,两幅关键画面之间需要插入多幅过渡画面。这种十分繁重的工作现在可以由计算机完成。交互式图形系统还可以用来设计一些十分复杂的特技镜头。可以设想,用计算机生产、复制、传输和存贮电影也是完全可能的,这对传统的电影生产、放影方式无疑会产生深远的影响。

总之,凡是用到图形、图象信息的场合,就有计算机图形系统的立足之地。而凡是需要人加以干预的场合,交互性就不可缺少。当然,在交互式图形系统的各种应用中,必须有各个具体部门的配合才行,包括各种测量仪表、模数与数模转换、各个过程的数学模型以及各个领域的专业知识,等等。

## 二、计算机图形学的研究内容

计算机图形学的主要任务是研究如何用计算机来处理图形。研究内容有:

### 1. 图形的输入技术

研究如何将要处理的图形输入计算机,以便计算机处理。

### 2. 图形的生成、显示和输出

研究如何利用图形在计算机内的表示方法在显示屏幕上显示图形和在绘图仪、打印机等输出设备上输出图形。

### 3. 图形的变换

图形的变换包括几何变换和色彩、灰度变换。

### 4. 图形的组合、分解

研究用简单的图形组成复杂的图形,或者将复杂的图形分解为简单的图形。

## 三、光栅图形扫描显示器基本原理

交互式图形处理系统的硬件除计算机主机外,主要是图形显示器。这一部分着重介绍图形显示器的结构和主要部件CRT(阴极射线管)的结构。

### 1. CRT 图形显示器

目前虽然已有各种各样的图形显示设备,但占统治地位的仍然是以 CRT 为核心部件的 CRT 显示器。

CRT 的结构见图 1.1,它主要由电子枪、偏转系统和荧光屏三部分组成。

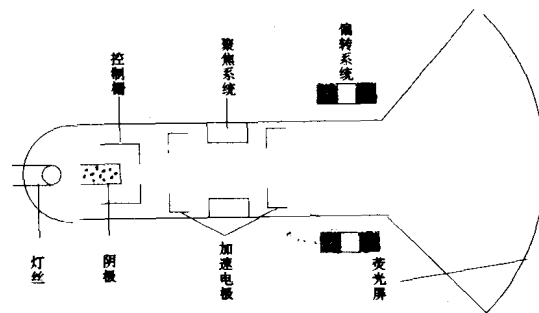


图 1.1 CRT 结构

#### 1) 电子枪

电子枪由灯丝、阴极和控制栅组成。电流通过灯丝,产生热量,即对阴极加热,从而使阴极发射出电子束。在聚焦极加上一定的正电压,使电子束聚焦,形成很细的电子束,再由加速极加上正电压对电子束加速,使它具有足够的能量射向荧光屏。靠近阴极有一个控制极,加上负电压后能控制电子束的强弱,甚至使电子束截止。也就是说,控制极能够控制电子束打在荧光屏上所产生的光的强弱。

#### 2) 偏转系统

静电场或电磁场可以用来控制电子束的偏转方向。通常利用两组偏转极或两组偏转线

圈来产生静电场或电磁场,使电子束作左右、上下偏转,从而控制荧光屏上光点的左右、上下运动。

### 3) 荧光屏

荧光屏上涂有荧光粉,电子束打在荧光屏上,荧光粉就会发光而形成光点。其颜色和余辉时间(电子束离开后继续发光的残留时间)的长短决定于荧光粉的种类。

单色 CRT 有一个电子枪,而彩色 CRT 一般有三个电子枪,电子枪和荧光屏之间的掩膜能控制不同电子枪发射的电子束射在荧光屏的不同部位,使不同的荧光粉发光,从而显示出不同的颜色。

荧光屏上显示的图形是由偏转系统控制电子束在荧光屏上运动产生的,也就是说,是电子束或光点“扫描”出来的。而光点的亮度则由控制极控制。扫描的方式有随机扫描和光栅扫描两种。

随机扫描方式与示波器的工作方式类似。比如,要产生一条从点 A 到点 B 的线段,是先把电子束从当前位置用暗矢量移到 A 点,然后用亮矢量把电子束从 A 点移到 B 点。

光栅扫描方式则完全不同。电子束受偏转部件的控制,不断从左到右、从上到下扫描整个屏幕。在这个过程中,只要控制电子束的强弱就能显示图形。整个屏幕由扫描线分成 m 行,每一行又分成 n 个小点,这样,整个屏幕就分为  $m \times n$  个小点,即像素。图形是电子束扫描到各个像素时产生不同亮度和颜色的光点形成的,这与电视机产生图象的方法类似。

随机扫描的优点是速度快,线条质量高,容易修改。光栅扫描则灰度和色彩丰富,能产生真实感很强的复杂图象。近十几年来,由于半导体存贮器性能越来越稳定可靠,价格也大幅度降低,使得光栅扫描图形显示器的价格也大幅度降低,其应用领域不断扩大,光栅扫描显示器已成为显示器市场的主流,成为个人计算机、图形工作站等各类计算机所使用的最重要的显示设备。

## 2. 光栅扫描显示器的结构与原理

图 1.2 表示的是光栅扫描显示器的基本结构。这里,显示存贮器是整个显示器的核心。显示存贮器又称位图存贮器、视频存贮器和帧缓冲区。

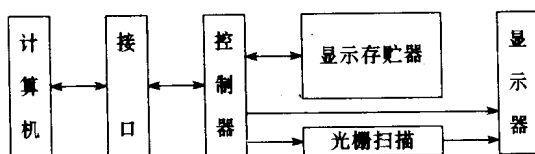


图 1.2 光栅扫描显示器

显示屏上的每一个像素对应显示存贮器中的若干位。对于单色图形,每个像素用一位就可以表示,位为 0 表示该像素为暗,位为 1 表示该像素为亮。如果每个像素用  $i$  位表示,则该像素有  $2^i$  级灰度,或者可以有  $2^i$  种颜色。显示存贮器中的这种信息称为位图。

计算机将要显示的图形、图象转化为位图,经过接口电路送入显示存贮器。而 CRT 控制器则依次扫描显示存贮器中的位图,将位图转换成亮度或色彩信号来控制电子束的强度。同时,CRT 控制器控制电子束自左向右、自上而下进行光栅扫描,这样就在屏幕上显示出所需的图形、图象。

为了得到稳定的图象,光栅扫描显示器要对显示存贮器重复扫描,进行刷新操作。屏幕刷新的速度一般为每秒 50 至 70 次。

评价显示器性能的一个重要指标是其分辨率。显示分辨率指屏幕水平方向可以显示的象素数目(水平分辨率)及垂直方向可以显示的扫描线数目(垂直分辨率)。高分辨率显示器可以显示的象素数目达一兆以上。

评价显示器性能的另一个重要指标是颜色数(或灰度等级数)。高性能彩色显示器可以同时显示  $2^{32}$  种颜色。

显示存贮器中存放的是被显示图形的位图,所以,显示分辨率越高,显示的颜色越多,所需要的存贮容量越大。显示存贮器一般用三维表示,它除了一般计算机内存的 X-Y 编址外,其地址对应的象素的亮度和颜色值也由每一位的深度来决定。比如,如果显示分辨率为  $320 \times 200$ ,显示颜色为 4 种,则显示存贮器要求有 16000 字节;而高分辨率的光栅显示器其显示分辨率可达  $1024 \times 768$ ,深度为 8,即可以显示 256 种颜色,这样,就至少需要 768K 字节的显示存贮器。

由于 CRT 控制器需要不断访问显示存贮器,读出位图,刷新屏幕上的画面,而在屏幕刷新的同时,计算机还需读写显示存贮器,以显示新的图形,所以,要求显示存贮器的工作速度很高。显示存贮器的速度是有限的,这就限制了显示存贮器的容量不能太大。

为了在不增加显示存贮器容量的条件下,增加显示器可以显示的颜色数目,可以使用调色板技术,也称颜色表技术。这是因为在大多数情况下,显示某一图形所需的颜色数目并不大,但对于不同图形,所需的总的颜色数目却很大。

使用颜色表时,显示存贮器中存放的不再是直接送往显示器的颜色值,而是要显示颜色的编号,简称颜色号。颜色表中存放的是各颜色号对应的 R、G、B 三原色的饱和度,这才是颜色值。CRT 控制器扫描显示存贮器时读出颜色号后,以此作为颜色表地址,经转换,得到颜色值,再送往显示器显示。如果显示存贮器的深度只有 8,那么颜色表就应该有  $2^8 = 256$  个地址,每个地址对应一个颜色号,当颜色表为 24 位,R、G、B 各用 8 位表示时,显示器可以显示的颜色数就可达  $2^{24} = 16M$  种。不过,由于显示存贮器深度为 8,所以,可同时显示的颜色数仍然为 256 种。颜色表的内容是可修改的,软件可以控制颜色表的内容,从 16M 种颜色值中选择出合适的 256 种颜色值,装入颜色表,使之符合当前图形显示的需要。显然,这样就提高了显示器图形输出的颜色范围,而显示存贮器的容量却没有增加。

## § 1.2 点生成算法

对于光栅扫描显示器,点(象素)是最基本的元素。但这里的点与几何中的点是有区别的,见图 1.3。几何的点只有位置而没有大小,而且点的坐标可以是任意的实数值。而所谓的象素点是有大小,且坐标常取整数值(对显示屏而言,常取正整数作为屏幕坐标)。以下就讨论如何用象素点来表示几何的线、圆和椭圆。

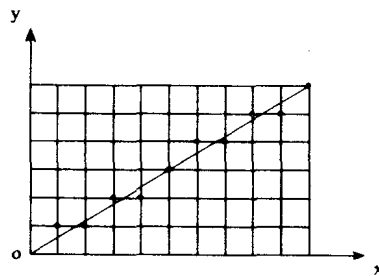


图 1.3 几何的直线与象素的关系

# 一、直线段的点生成算法

## 1. 逐点计算法

假定要画由 $(x_1, y_1)$ 到 $(x_2, y_2)$ 的一个线段,可写出它的参数方程:

$$x = x_1 + (x_2 - x_1)t$$

$$y = y_1 + (y_2 - y_1)t$$

$$(0 \leq t \leq 1)$$

$$\text{令 } t = t_i = \frac{i}{n}$$

$$(i = 0, 1, \dots, n)$$

则可以得到点列:

$$x_i = x_1 + (x_2 - x_1)t_i$$

$$y_i = y_1 + (y_2 - y_1)t_i$$

$$(i = 0, 1, \dots, n)$$

$M_i = ([x_i], [y_i])$ 就是象素点,  $n+1$  个点可以表现直线段。为了使象素的密度不因线段的长度而变,  $n$  自然应随长度而定, 例如取  $n = [\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}]$ 。为了简化亦可以取  $n = \max(|x_2 - x_1|, |y_2 - y_1|)$ 。

用这种方法, 每画出一个点就要作一次乘法和加法, 由于乘法很费时间, 所以此算法的速度不快。下面的 DDA 算法对此作了改进。

## 2. DDA 方法产生直线段

DDA 算法是建立在图形元素所满足的微分方程的基础上, 由 $(x_1, y_1)$ 到 $(x_2, y_2)$ 的直线段所满足的微分方程为

$$\frac{dx}{dt} = x_2 - x_1$$

$$\frac{dy}{dt} = y_2 - y_1$$

从而

$$\frac{\Delta x}{\Delta t} = x_2 - x_1$$

$$\frac{\Delta y}{\Delta t} = y_2 - y_1$$

或

$$\Delta x = (x_2 - x_1)\Delta t$$

$$\Delta y = (y_2 - y_1)\Delta t$$

令 $\Delta t$  满足 $\Delta t = \frac{1}{L}$ ,  $L = \max(|x_2 - x_1|, |y_2 - y_1|)$ , 计算出 $\Delta x, \Delta y$ 。

$$x_{i+1} = x_i + \Delta x$$

$$y_{i+1} = y_i + \Delta y$$

由 $(x_i, y_i)$ 可得 $(x_{i+1}, y_{i+1})$ 。除了开始计算 $\Delta x, \Delta y, \Delta t$ 时要乘除法外, 以后每次只要做

加法就行了。输出的点是 $([x_{i+1}], [y_{i+1}])$ 。

### 3. Bresenham 算法

由于直线段是形成更复杂图形元素的基础,因而对生成直线段算法的任何改进,意义都很重大。Bresenham 于 1965 年提出的算法对 DDA 算法作了改进。它的基本思想是:每步一个坐标改变 $\pm 1$ ,而另一个坐标要看当前点与直线上点的误差 $e$ , $e$ 的符号决定了是否改变另一个坐标。这个误差项 $e$ 记录的是在同最大位移垂直的方向上,直线的精确路径同实际产生的点之间的距离。如图 1.4 所示, $x$ 轴是最大的位移方向,误差项 $e$ 表明的是在同 $y$ 轴平行的方向上算出的直线精确路径同实际产生的点之间的距离。以下讨论按照这个约定进行。

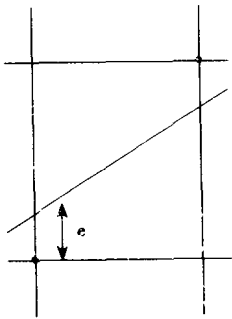


图 1.4 误差  $e$

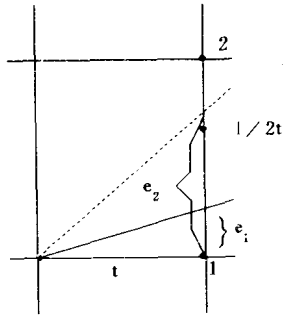


图 1.5 误差  $e$  对走步的影响

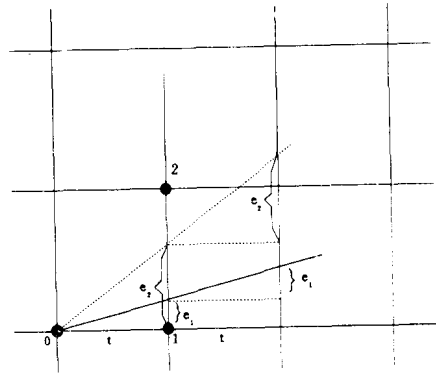


图 1.6 误差  $e$  的计算

如图 1.5 所示,设步长为  $t$ ,对于图中实线, $e_1$  应该等于  $\frac{\Delta y}{\Delta x}t$ ,因  $e_1 < \frac{1}{2}t$ ,直线的精确路径更接近点 1,故第一步应显示点 1,即  $y$  方向坐标不变;同理,对于图中虚线, $e_2 = \frac{\Delta y}{\Delta x}t$ ,而  $e_2 > \frac{1}{2}t$ ,则应显示点 2。如果我们让  $e$  减去  $\frac{1}{2}t$ ,即  $e = \frac{\Delta y}{\Delta x}t - \frac{1}{2}t$ ,此时若  $e \leq 0$ ,则  $y$  方向不变,若  $e > 0$ ,则  $y$  方向坐标加上  $t$ ,即增加一步。

下一步误差的计算应该分两种情况来讨论。先看图 1.6 中的实线,即  $y$  方向没有走步的情形。很明显,此时直线的精确路径与已经产生的点 1 的误差又增加了  $e_1$ ,则新的误差项应该是  $e_1 + e_1 - \frac{1}{2}t$ ,即原先所保留的  $e$  再加上  $e_1$ ,即为  $e + \frac{\Delta y}{\Delta x}t$ 。

再看图 1.6 中虚线,即  $y$  方向增加一步的情形,同实线的情形相似,在原来误差上又增加  $e_2$ ,但由于  $y$  方向还增加了一步,即误差的值应该是又缩小了一个步距  $t$ ,即为:

$$e - \frac{\Delta y}{\Delta x}t - t。$$

按照这种判别规则,用程序实现起来是十分容易的。但是我们注意到在  $e$  的计算中要多次用到除法,这在实际实现起来比较复杂,而且很费时。然而我们发现,在程序中仅仅是  $e$  的符号被用来判定  $y$  方向的走步情况,而  $e$  的数值无关紧要。这样,当我们给  $e$  乘上  $\frac{2\Delta x}{t}$  后,并不影响判断的正确性,而判别式却可以大大化简。如对于初始的  $e$  值:

$$e = \frac{\Delta y}{\Delta x}t - \frac{t}{2} = 2 \cdot \Delta y \cdot t - \Delta x \cdot \frac{t}{2\Delta x}$$

等式两边乘以  $2 \frac{\Delta x}{t}$  后,则初值



$$e = 2\Delta y - \Delta x$$

对于 y 方向不走步这种情况, 下一步的误差项则可简化为:  $e + 2\Delta y$ 。对于 y 方向走一步这种情况, 下一步的误差项则可简化为:  $e + (2\Delta y - 2\Delta x)$ 。由于一个数乘以一个常数 2, 在计算机上可以通过简单的移位来实现, 所以实现上述判断的逻辑十分简单。

除了上述  $0 \leq \Delta y \leq \Delta x$  的情况外, 其它的情况对于 Bresenham 方法也同样可以实现, 只是误差项的计算要稍作修改。

从端点 1( $x_1, y_1$ ) 至端点 2( $x_2, y_2$ ) 的任意直线的扫描转换算法流程如下:

1)  $dx = |x_2 - x_1|, dy = |y_2 - y_1|, x = x_1, y = y_1, i = 0$ ; 如果  $x_2 \geq x_1$ , 则  $incx = 1$ , 否则  $incx = -1$ ; 如果  $y_2 \geq y_1$ , 则  $incy = 1$ , 否则  $incy = -1$ ;

2) 如果  $dx > dy$ , 转 7;

3)  $e = 2dx - dy, inc1 = 2dx - 2dy, inc2 = 2dx$ ;

4) 写点  $(x, y), y = y + incy$ ;

5) 如果  $e < 0, e = e + inc2$ ; 否则  $x = x + incx, e = e + inc1$ ;

6) 如果  $i = dy$ , 转 11; 否则  $i = i + 1$ , 转 4;

7)  $e = 2dy - dx, inc1 = 2dy - 2dx, inc2 = 2dy$ ;

8) 写点  $(x, y), x = x + incx$ ;

9) 如果  $e < 0, e = e + inc2$ ; 否则  $y = y + incy, e = e + inc1$ ;

10) 如果  $i = dx$ , 转 11; 否则  $i = i + 1$ , 转 8;

11) 退出。

由于 Bresenham 算法避免了乘法和除法, 因此它能够更好地适应应用硬件或者微处理机来执行。该算法不仅速度快, 而且逼近效果也是最好的。

## 二、圆弧、椭圆弧的点生成算法

### 1. 圆弧的 DDA 算法

为了描述的简单, 这里先假定圆心在原点。这时, 圆弧满足的参数方程为  $x = R \cos t, y = R \sin t$ 。从而得出

$$dx = -R \sin t dt = -y dt$$

$$dy = R \cos t dt = x dt$$

于是可以用下面的差分方程来近似代替, 即

$$\Delta x = -y \Delta t$$

$$\Delta y = x \Delta t$$

若点  $(x_i, y_i)$  在圆弧上, 令  $\Delta t = \delta$ , 则

$$x_{i+1} = x_i - \delta y_i$$

$$y_{i+1} = y_i + \delta x_i$$

而  $(x_{i+1}, y_{i+1})$  近似地在圆弧上。为了使  $(x_{i+1}, y_{i+1})$  与  $(x_i, y_i)$  邻近, 应该使

$$(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 = \delta^2 (y_i^2 + x_i^2) = \delta^2 R^2 \leq 2$$

当  $2^{(n-1)} \leq \frac{R}{\sqrt{2}} \leq 2^n$  时, 可取  $\delta = 2^{-n}$ 。