

HOPE 适用于 IBM PC · PS/2 及其兼容机

微机图形开发与动画技术



北京希望电脑公司

适用于 IBM PC、PS / 2 及其兼容机

微机图形开发与动画技术

邹然军 编译

北京希望电脑公司

一九九二年二月

目 录

前 言

第一章 计算机图形的组成元素	(1)
1.0 概述	(1)
1.0.1 图形语言	(1)
1.0.2 图象感知	(1)
1.0.3 形象思维	(2)
1.1 技术元素	(3)
1.1.1 主要图形硬件	(4)
1.1.1.1 存贮管理显示器	(4)
1.1.1.2 向量刷新显示器	(4)
1.1.1.3 光栅扫描显示器	(4)
1.1.2 图象的物理性质	(7)
1.1.2.1 亮度和对比度	(7)
1.1.2.2 颜色	(8)
1.1.2.3 分辨率	(8)
1.1.2.4 长宽比	(8)
1.1.3 访问显示元素	(8)
1.1.3.1 多路复用与分时	(9)
1.1.3.2 内存映象	(9)
1.2 人的因素	(10)
1.2.1 视觉器官	(10)
1.2.2 视觉调节	(11)
1.2.3 眼睛运动	(11)
1.2.4 滞留	(11)
1.2.5 敏锐度	(12)
1.2.6 视觉范围	(12)
1.2.7 眼睛疲劳	(12)
1.2.7.1 催眠效果	(13)
1.2.7.2 图形组合与眼睛疲劳	(13)
1.3 美学元素	(13)
1.3.1 图形组合与图形美	(13)
1.3.1.1 平衡和重量	(14)
1.3.1.2 形状 (shape) 和形态 (Form)	(15)
1.3.1.3 图和背景	(15)

1.3.1.4	空间和深度感	(16)
1.3.1.5	光与色	(19)
1.3.2	文本字符的美学	(19)
第二章	IBM 显示器和图形资源	(20)
2.0	IBM 图形硬件	(20)
2.0.1	图形卡和图形系统	(21)
2.0.2	字母数字方式和图形方式	(21)
2.0.3	显示存储器	(23)
2.1	PC 显示系统	(24)
2.1.1	单色显示适配器 (MDA)	(24)
2.1.2	Hercules 图形卡 (HGC)	(26)
2.1.3	彩色图形适配器 (CGA)	(26)
2.1.4	增强图形适配器 (EGA)	(27)
2.1.5	PCjr 显示硬件	(28)
2.2	PS/2 显示系统	(30)
2.2.1	MCGA 系统	(30)
2.2.2	VGA 系统	(31)
2.2.3	PS/2 系统监视器	(31)
2.3	高分辨率图形板	(31)
2.3.1	IBM 专用图形系统	(32)
2.3.2	8514/A 显示适配器	(32)
第三章	图形软件	(34)
3.0	图形程序设计	(34)
3.0.1	独立于设备的图形学	(35)
3.1	虚拟图形机器	(36)
3.1.1	虚拟图形设备	(36)
3.1.2	输出函数 (图元)	(37)
3.1.3	输入函数	(37)
3.1.4	显示文件	(39)
3.1.4.1	坐标标准化	(40)
3.1.4.2	显示文件命令	(40)
3.2	图形软件标准	(40)
3.2.1	图形核心系统 (GKS)	(41)
3.2.1.1	GKS 工作站	(42)
3.2.1.2	GKS 坐标系统	(42)
3.2.1.3	GKS 输出图元	(43)
3.2.2	虚拟设备接口 (VDI)	(45)
3.2.3	虚拟设备元文件 (VDM)	(45)
3.2.4	标准化组织	(45)

3.2.5	标准化与软件开发	(45)
3.3	高级语言图形学	(46)
3.3.1	图形模型	(46)
3.3.2	传统语言图形学	(47)
3.3.3	系统软件对图形的支持	(47)
3.3.3.1	BIOS 支持	(48)
3.3.3.2	OS / 2 对图形的支持	(48)
3.3.4	图形语言	(48)
3.3.4.1	PostScript	(49)
3.3.4.2	PHIGS	(49)
3.4	机器级图形学	(50)
3.4.1	设备驱动器	(50)
3.4.2	图元	(50)
3.4.3	低级语言形式的图形应用程序	(51)
3.5	图形开发工具	(51)
3.5.1	混合语言图形程序	(51)
3.5.2	用 C 语言设计图形	(52)
第四章	图形程序设计工具	(53)
4.0	图形程序开发	(53)
4.1	汇编语言程序设计工具	(53)
4.1.1	编辑程序	(54)
4.1.2	汇编程序	(54)
4.1.2.1	宏汇编程序	(55)
4.1.3	连接程序	(55)
4.1.4	用软件仿真 8087	(56)
4.1.5	调试程序	(57)
4.2	高级语言处理程序	(57)
4.3	图形和数学协处理器	(58)
4.3.1	8087 数学协处理器	(58)
4.3.2	8087 芯片简介	(58)
4.3.3	8087 用于图形程序设计	(59)
4.3.4	8086 / 8087 接口与同步	(61)
4.3.5	8087 体系结构	(61)
4.3.5.1	8087 寄存器栈	(62)
4.3.5.2	8087 控制寄存器	(62)
4.3.5.3	8087 状态寄存器	(65)
4.3.5.4	指令和数据指针	(65)
4.3.6	8087 的程序设计	(65)
4.3.6.1	使用 8087 的整数传送	(67)

4.3.6.2	8087 寻址方式	(68)
4.3.6.3	8087 编码实例	(69)
4.3.6.4	异常处理	(70)
4.3.7	8087 指令集	(70)
4.3.7.1	数据传送指令	(71)
4.3.7.2	算术指令	(71)
4.3.7.3	比较指令	(72)
4.3.7.4	超越指令	(73)
4.3.7.5	常量指令	(73)
4.3.7.6	处理器控制指令	(74)
4.3.8	8087 软件支持	(75)
4.3.9	三角函数的计算	(76)
第五章	逻辑与数学程序元素	(80)
5.0	图形结构	(80)
5.1	图象	(80)
5.1.1	常用几何图象元素	(80)
5.1.1.1	点	(81)
5.1.1.2	线段	(81)
5.1.1.3	曲线和弧线	(81)
5.1.1.4	多边形	(82)
5.1.1.5	编码单元	(83)
5.1.2	图象映象	(83)
5.1.2.1	显示缓冲区	(83)
5.1.2.2	图象缓冲区	(84)
5.1.2.3	观察口	(85)
5.1.2.4	窗口	(85)
5.2	显示文件	(85)
5.2.1	图象文件	(86)
5.2.2	图象段	(87)
5.2.3	图象描述符	(87)
5.3	段操作	(87)
5.3.1	命名与创建段	(88)
5.3.2	打开和关闭段	(88)
5.3.3	更名和删除段	(88)
5.4	段属性	(88)
5.4.1	可视性	(89)
5.4.2	线颜色、填充颜色和线类型	(89)
5.4.3	优先级	(89)
5.4.4	图象变换	(89)

5.5	图形数据文件	(90)
5.5.1	编码显示文件	(90)
5.5.1.1	描述符编码	(90)
5.5.1.2	段文件编码	(91)
5.5.1.3	图象文件编码	(92)
5.6	图象变换工具	(93)
5.6.1	坐标矩阵	(93)
5.6.2	矩阵算术运算	(94)
5.6.2.1	标量乘矩阵运算	(94)
5.6.2.2	矩阵加减运算	(95)
5.6.2.3	矩阵乘运算	(95)
5.7	几何变换	(96)
5.7.1	平移	(96)
5.7.2	比例变换	(97)
5.7.3	旋转	(98)
5.7.4	齐次坐标	(100)
5.7.5	连接	(101)
5.8	通过图象变换产生的动画	(102)
5.8.1	简单动画	(103)
5.8.2	复杂动画	(104)
第六章	图形硬件的程序设计	(104)
6.0	IBM 微机上的图形程序设计	(104)
6.0.1	选择一个程序设计模型	(105)
6.0.1.1	兼容性	(105)
6.0.1.2	对称分辨率	(106)
6.0.1.3	在 EGA 系统中执行实例程序	(106)
6.0.2	EGA 和 VGA 结构	(107)
6.0.3	图形方式	(108)
6.0.4	显示存贮映象	(108)
6.0.4.1	显示地址压缩	(108)
6.0.4.2	锁存寄存器	(108)
6.0.4.3	存贮映象	(108)
6.0.4.4	访问位平面 (BitPlane)	(109)
6.1	预备操作	(111)
6.1.1	显示系统标识	(111)
6.1.2	GRAPHSYS 程序	(112)
6.1.3	BIOS 数据区	(118)
6.1.4	DOS 下显示方式的设置	(119)
6.1.5	DOS 下显示缓冲区地址的设置	(119)

6.2	可编程显示器件	(120)
6.2.1	EGA 和 VGA 图形显示器件	(120)
6.2.1.1	通用寄存器	(122)
6.2.1.2	CRT 控制器	(122)
6.2.1.3	时序发生器	(122)
6.2.1.4	图形控制器	(125)
6.2.1.5	属性控制器	(131)
6.2.2	数—模转换器 (DAC)	(135)
6.3	DOS 下的图形软件	(137)
6.3.1	BIOS 提供图形服务	(137)
6.3.2	扩充的 BIOS 显示服务	(137)
6.3.2.1	BIOS 的扩充服务号 16	(137)
6.3.2.2	BIOS 的扩充服务号 17	(138)
6.3.2.3	BIOS 扩充的服务号 18	(140)
6.3.2.4	BIO 扩充的服务号 26	(141)
第七章	OS / 2 环境下的图形程序设计	(142)
7.0	多任务和图形程序设计	(142)
7.0.1	存贮器的限制	(142)
7.0.2	特权级	(143)
7.0.3	输入和输出的限制	(143)
7.0.4	资源共享	(144)
7.0.4.1	线索 (Thread)	(144)
7.0.4.2	进程 (Process)	(145)
7.0.4.3	会话 (Session)	(145)
7.1	OS / 2 程序开发	(146)
7.1.1	段结构	(146)
7.1.1.1	双模态 (Bimodal) 执行	(148)
7.1.1.2	8086 和 8088 的兼容性	(149)
7.1.2	创建 IOPL 段	(150)
7.1.3	连接时的要求	(150)
7.1.4	执行特权代码	(151)
7.2	OS / 2 中的准备操作	(151)
7.2.1	OS / 2 中显示系统的识别	(151)
7.2.2	OS / 2 中显示方式的设置	(154)
7.2.3	取得对物理缓冲区的控制	(156)
7.2.4	屏幕的封闭与开启	(157)
7.2.5	结束图形对话	(158)
7.3	OS / 2 图形方式下的字符显示	(158)
7.3.1	将 ROM 中的字体表装入 RAM	(159)

7.4	多任务图形程序	(160)
7.4.1	OS / 2DOS 方式下的图形应用程序	(161)
7.4.2	OS / 2 保护方式下的图形应用程序	(161)
7.4.3	存贮—重画—等待线索	(162)
7.4.4	关键部分	(165)
7.4.5	图象的存贮与恢复	(165)
7.5	作为后台任务的图形输出	(165)
7.5.1	到虚拟设备的图形输出	(166)
7.5.2	VGA 方式 18: 虚拟设备驱动程序	(166)
第八章	开发设备驱动程序	(170)
8.0	设备驱动程序和图元	(170)
8.0.1	前提条件和准备操作	(171)
8.0.2	显示彩色映象和象元模式	(171)
8.1	EGA 和 VGA 写方式	(172)
8.1.1	选择写方式	(173)
8.1.2	EGA 和 VGA 写方式 0	(174)
8.1.3	EGA 和 VGA 写方式 1	(175)
8.1.4	EGA 和 VGA 写方式 2	(176)
8.1.5	VGA 写方式 3	(177)
8.2	VGA 和 VGA 读方式	(177)
8.2.1	选择读方式	(177)
8.2.2	EGA 和 VGA 读方式 0	(178)
8.2.3	EGA 和 VGA 读方式 1	(178)
8.3	VGA 方式号 18: 设备驱动例程	(178)
8.3.1	地址计算	(179)
8.3.1.1	粗纹理地址计算	(180)
8.3.1.2	细纹理地址计算	(180)
8.3.2	通用象元写例程	(182)
8.3.3	字节级读例程	(183)
8.3.4	象元级读例程	(184)
8.4	图形方式打印机驱动程序	(185)
8.4.1	控制打印头	(185)
第九章	开发图元	(186)
9.0	图形例程	(186)
9.1	粗纹理图元	(186)
9.1.1	粗边界矩形区域填充例程	(187)
9.1.2	粗边界位模式显示例程	(189)
9.2	细纹理图元	(190)
9.2.1	细边界区域填充例程	(190)

9.2.2	细边界位模式显示例程	(191)
9.2.3	内存中位模式的编码	(193)
9.3	直线几何图元	(193)
9.3.1	直线的象元表示法	(194)
9.3.1.1	近似性	(194)
9.3.1.2	相邻性	(194)
9.3.2	修改不相邻图	(195)
9.3.3	不相邻图的修改例程	(198)
9.3.4	点到点的画线例程	(199)
9.3.5	点和斜率的画线例程	(200)
9.3.6	显示和存贮坐标	(202)
9.4	二次曲线几何图元	(203)
9.4.1	画圆例程	(203)
9.4.2	画椭圆例程	(204)
9.4.3	画抛物线的例程	(206)
9.4.4	画双曲线的例程	(207)
9.4.5	扇形的显示例程	(209)
9.5	文本显示图元	(209)
9.5.1	字符发生器例程	(210)
9.5.2	字符发生器图元	(210)
9.6	用于图形动画的图元	(212)
9.6.1	实时动画	(212)
9.6.2	交互式映象例程	(213)
9.6.2.1	击键拦截例程	(213)
9.6.2.2	交互式屏幕显示	(215)
9.6.3	时钟脉冲动画	(217)
9.6.3.1	DOS 下的时钟脉冲动画	(217)
9.6.3.2	OS/2 下的时钟脉冲动画	(219)
9.7	其它图元	(219)
9.7.1	打印机控制例程	(219)
9.7.2	向量命令例程	(221)
9.8	高级语言接口	(222)
9.8.1	C 语言接口	(223)
9.8.1.1	C 调用汇编模块	(223)
9.8.1.2	接口首部 (Heading) 的实例	(223)
附录 A	/2FLAG.ASM 程序	()
B	CONICS.ASM 程序	()
C	OKIDUMP.ASM 程序	()
D	ANIMATE.ASM 程序	()

第一章 计算机图形的组成元素

1.0 概述

一幅图 (graph) 就是一幅图象 (picture)。计算机图形学指的是一项技术, 有了它便可以利用计算机创建和管理图象。在遇到“计算机图形学”这个术语时, 不同的读者会联想到不同的应用领域, 例如:

1. 带有动画、多种彩色显示和用于图象控制及音响效果的机械装置的快速电子连拱廊游戏。
2. 计算机辅助设计 (CAD) 已取代了许多工程师和建筑师们原先使用的丁字尺和机械式绘图机器。
3. 计算机化的飞行模拟程序非常逼真, 可用来训练飞行员。
4. 人体内部结构和器官的扫描设备, 可用于医疗诊断系统。
5. 电子印刷系统能够进行绘画、速描甚至照像, 并带有各种几何设计和文本字体。

但是, 上面列举的各项只不过是当前计算机图形应用的一斑而已。其它许多计算机控制的图形系统用于家庭、商业和高等学府, 应用领域涉及到艺术、科学、金融、教育和技术。它们的应用目的可以是严格面向科学的和纯粹面向艺术的。计算机的图形硬件从 Atari 到专用的大型机应有尽有。

1.0.1 图形语言

人脑最具魅力的能力便是能够从思想中提取基本元素并将这些元素体现在简化的图象模式中。人脑还能够从抽象图画 (schematic depiction) 推测出原意来。这也是值得称道的。奔鹿符号是美国许多乡村公路常见的路标。

对普通司机而言, 该图传递了几种信息: (1) 该地区有野鹿; (2) 这些动物可能会穿过道路; (3) 务必小心驾驶以免撞上。要想用文字来表达奔鹿图中如此具有说服力而又简捷的信息是困难的。即使可以找到文字来表达, 但图形方式仍具有这样的优点, 即: 不懂得该文字的驾驶员也看得懂。

从我们的穴居祖先们开始在墙壁上画图开始, 图形便成了表达思想的一种非语言手段。实际上, 现在大家已公认, 书面语言起源于所熟悉语言的图形。这些图形符号后经演变、简化就成了书面的文字。譬如, 在埃及的象形文字中, 许多物体今天还可以辨认出来。在现代的日文和中文汉字中这些图象的起源是显而易见的。

1.0.2 图象感知

感知 (perception) 是人注意到事物的一个过程。多少个世纪以来, 哲学家们一直在争论信息获取和交换过程中所产生的肉体过程和精神变化。有些研究涉及到代表客观存在的文字、图形和模型的生成与理解的精神过程。毫无疑问, 这种第二性感知形式的最重要表达方式便是语言。已有不少文章对语文做过论述。语言对于文化是极为重要的, 所以我们花费了大量时间来接受语言教育。不过, 图象和符号也被广泛用来获取和传达信息。事实上, 它们在传达信息方面往往是最有效的手段, 然而, 在有关图象和符号的生成与理解

方面，我们一般均未受过正规训练。在这方面，我们主要凭直觉（intuition）来利用图象符号并希望它们传递想要传递的信息。



Figure 1-1.

图 1-1 美国乡村公路上的奔鹿路标

历史上对感知机制的一个解释是以思想存在并能够传达作为前提的。根据该理论，符号内部便带有思想。但是，对感知的这种直观解释引入的概念与它打算解释的对象一样深奥。Jame J. Gibson 在这方面提出一个较为满意的解释。他把符号解释为另一个人产生的象征性刺激体（surrogate stimulus）。他认为人类倾向于寻找象征物，象征物的形式可以是噪音、身体动作、图形、绘画、书写、雕刻、造型甚至音乐声。

应用这些符号的最重要结果可能是让别人能够理解创造符号的人观察到的事件和对象。请注意这样一个有趣的现象，即：创造符号的这种过程涉及到自刺激（self-stimulation）。说话者听到自己的声音并为之高兴，艺术家在自己的素描、油画或雕刻作品中找到了快乐，作家在自己的诗歌或散文中发现了喜悦，作曲家在自己的音乐中得到了快慰。创造符号的人同时又是观察符号的人。他们对自己符号的价值往往不能够作出正确判断。

1.0.3 形象思维

一九四六年，Charles W. Morris 在其作《符号、语文和行为》中指出，在符号的创造者和阅读者之间的循环反应成功地解释了在没有第二个人甚至没有刺激对象时，噪音是如何转化成符号的，而符号的创造和阅读周期又是如何发生的。儿童的这一发展阶段与思维阶段的开始是一致的。由于这个循环方式可以存在于非语言符号中，这样的假说是合乎情理的：思维过程发生在图象以及文字或其它知识符号方面。

过去大家往往以下例来示范形象思维：描述计算一天中时刻的两条思维路径。例如，如果在十一点五十分要计算一个半小时以后的时间，可如下进行算术运算：

$$11'50'' + 30'' = 11'80'' = 11' + 1' + (80'' - 60'') = 12'20''$$

我们也可以想象钟的表面，在想象中将钟表面一分为二，如图 1-2 所示。

在第一种情况下，是按照数值关系得到答案的，可能几乎没用到外部图象。在第二种情况下，想象的钟表面图案是获得答案的手段。有人发现，数字钟作为计时设备没有传统的带有刻度的钟令人满意，因为数字钟与我们想象的时间图象没有关系。

在传达知识概念时，并非一定能够找到方便且有关的图象。概念越抽象，越难找到相关的图象。例如，动物界包括如此众多相貌和特点不同的动物种类，所以实际上找不到一个有意义的图形符号将它们全部归纳起来。几百中的几个成员也是不能代表的。然而，在我们的思维中，可以采用动物王国的概念和动物种类中几百万个体的概念。

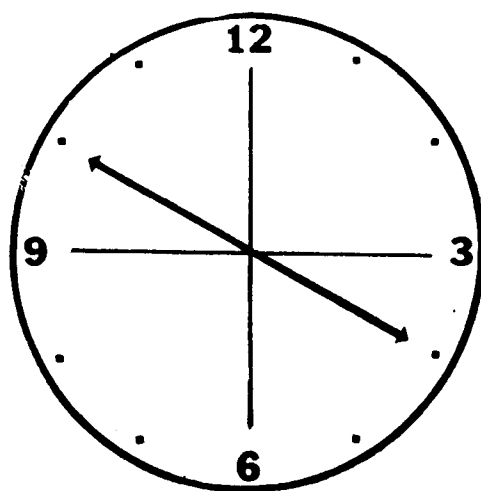


Figure 1-2.

图 1-2

这个事实使许多西方作家得出这样一个结论，即：思维的发生独立于感知经验，也就是说，思维只按照文字和数字发生。在十八、十九甚至二十世纪的一部分时间里，感知经验常被视为色欲的，甚至有罪的，而想象的图象则被看作纯属感官享受。这段时间的数学书中也很少有图画和插图。同时，数学家们认为只能偶尔采用速描和绘画来帮助说明特别的地方，因为数学真理和示范得靠理解力去掌握。

现在，我们越来越注意到图象在思想的产生和交流中所起到的重要作用。我们鼓励儿童用图象表达思想，同时也不再认为微分书中的插图不体面的了。但是，我们还远没有理解图象在思维过程中起到的作用。我们知道自己有固有的能力借助图形表达思想，但由于并不真正了解其中的机制，所以主要靠本能和直觉来创造这些图象。

最初的印象派画家演变成了现代画家。后者无疑是一切时代中最伟大的图形学探索者。他们的成果已改变了我们大部分的美学和感知概念。但是，这项革命一直不为重视。

1.1 技术元素

计算机并不是第一个将阴极射线管用作图形显示器的机器。示波器是实验室的一种电子设备，它实际上管理和计算输入信号以便将电波或电子波的图形显示在荧光屏上。因此，示波器是一种原始图形机器。它对计算机图形的主要贡献在于它证实了采用 CRT（阴极射线管）来显示用电子方式生成的图象是可行的。

在交互式图形系统中，用户可以创建或修改显示在 CRT 上的图象。在五十年代，开发了第一批军用交互式计算机图形设备。Sage 战略防空系统是一个多用户图形系统，它采用光笔生成的反馈信息。光笔是一种输入设备，用它通过计算机屏幕或分离的光敏表面将信息送入系统。在六十年代初期，Ivan Sutherland 和他的同事们在 M.I.Lincoln 实验室

为 Tx-2 计算机开发出了名为 SKETCHPAD 的图形的系统。该系统将光笔用作交互式图形设备。SKETCHPAD 的许多原理和操作变成了后来图形系统的标准。

尽管本书的目的主要是讨论 ICM 微型机中的图形系统和设备,但下面还是对计算机图形硬件做梗概性叙述。所叙述的某些系统和设备是具有历史意义的,这方面的叙述是简捷的。其它的叙述与现在正在使用的图形系统有关,届时将提供有关的系统名称及规格。

1.1.1 主要图形硬件

在过去的十年中,图形设备可以说已是品种繁多了。绘图仪、传统打印机、激光打印机、光笔、数字化面板、跟踪球定向设备和若干种 CRT 显示器都是图形硬件。但主要图形设备和辅助图形设备是有区别的。主设备参预原始图象的生成和显示,包括 CRT、图形控制设备,某些系统也包括存贮区。辅助图形设备主要用于图形输入和输出,如键盘、指向设备、数字化面板、打印机和绘图仪。在下面章节中,我们仅限于讨论主要图形器件,如 CRT 和 CRT 支持硬件。特殊的辅助图形设备在本书其它章节中讨论。

CRT 显示器最常用的技术可如下分类:

- a) 存贮管显示器
- b) 向量刷新显示器
- c) 光栅扫描显示器

1.1.1.1 存贮管显示器

从硬件角度看,这是最简单的 CRT 显示器。管表面涂有专门配制的磷粉,受到电子束撞击后可持续发出荧光达一个多小时。因此,阴极射线管既可用作显示器又可用作存贮设备并由此而得名。要想擦除整个显示器,应给管子加上电压,电压可将磷转变成黑色状态。对于这种操作方式,要擦除屏幕的特定区域或行,而同时保留其它部分是不可能的。可见,在这类系统中,图形动画是无法实现的。

存贮管显示器的其它缺点是:(1)如改变了一个显示元素,其它所有图象也要重画;(2)对比度低;(3)系统无彩色功能。因此,与其它型 CRT 显示器相比,与存贮管系统打交道既慢又难。由于这样一些因素,存贮管显示器虽然既简单又能持续显示,但现代图形终端极少采用,而微机图形系统中更是从来没有应用。

1.1.1.2 向量刷新显示器

与存贮管显示器相比,向量刷新显示器所使用的磷粉持续显示时间较短,必须每秒重新点亮 30—50 次。图 1-3 中示出了简化的向量刷新 CRT 的工作原理。

除具有阴极射线管之外,向量刷新系统还要求具有一个显示文件和显示控制器。显示文件是一片存贮区,它存放了绘制待显示目标的指令。显示控制器从显示文件中读取该信息,然后转换成对 CRT 电子元件的控制操作。

向量刷新显示器与存贮管显示器相比有若干优点,突出的有:(1)选择性擦除屏幕元素;(2)与操作员有高度交互性;(3)具有平移、比例变换、旋转变换等图象变换功能。向量刷新 CRT 的主要缺点是造价昂贵而且彩色功能不够强。

1.1.1.3 光栅扫描显示器

直到六十年代末,市场上可以买到的图形设备只有既昂贵而功能又弱的存贮管和向量刷新 CRT。从此以后,电视技术的重大突破使我们可以采用批量生产的元器件作为计算机系统的显示设备。Conrac 公司在这方面做出了开拓性的研究。该公司利用电视接收器

使用的图像刷新方法以及其它电视标准、设备和概念开发了一项计算机图象处理新技术，称为光栅扫描图形学。

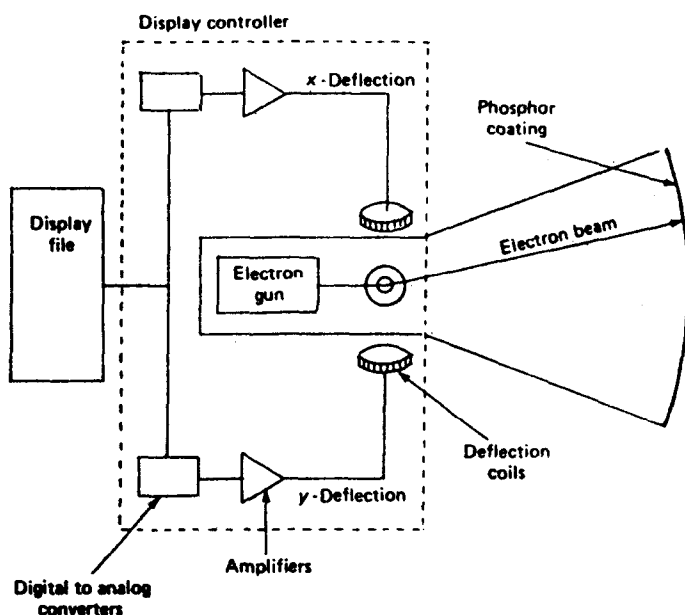


图 1-3 CRT 操作

图 1-4 中示出了光栅扫描显示器电子束经过的路径。扫描周期为每秒钟 50-70 次。在每条水平线开始处，控制器打开电子束，而在水平与垂直回扫周期里，电子束关闭。

光栅扫描 CRT 的显示器表面按物理划分成一个个由点 (dot) 组成的阵列，一般称之为象元 (pixels，该词由 picture 和 elements 而得)。IBM 采用术语 PEL 来表示屏幕上一个个的点。通常在计算机系统的内存空间中保留一片 RAM 区来记录每个屏幕象元的状态。最简单可行的存贮模式对应于黑白显示系统。对这种系统，每个象元只需要屏幕存贮区的一个位 (bit)，如果某位被设置，那么显示器的扫描器便点亮相应的象元。如果该位清除，那么相应的象元便会变黑 (不亮)。为屏幕显示器保留的内存区常被称为帧缓冲器 (frame buffer) 或视频缓冲器 (video buffer)，而整个显示系统则叫作内存映象的。图 1-5 中示了一个内存映象的黑白显示器的基本元素。

内存映象系统若用在彩色显示器上则需要更加复杂的模式：要给 CRT 配备若干个电子枪，每种颜色用一支枪激活象元。在以下的叙述中，假定一个彩色系统带有红、绿和蓝色成份。在这种情形下，需要用三个分开的内存映象来存放每个象元的状态。内存映象 R 反映象元的红色成份的状态。内存映象 G 和 B 存放存绿色和蓝色成份的状态。如果某象元的 R、G 和 B 位均处于设置态，则所有三个彩色枪都用来激活它，那么象元将是白色的。如果只有 R 和 G 置位，则象元被红和绿枪点亮，结果呈黄色。表 1-1 示出了红、绿、蓝三种颜色可能出现的组合。

表 1.1 红、绿、蓝三色的颜色组合

编号	红	绿	蓝	颜色
1	0	0	0	→黑
2	1	0	0	→红
3	0	1	0	→绿
4	0	0	1	→蓝
5	1	1	0	→黄 (非蓝)
6	1	0	1	→品红 (非绿)
7	0	1	1	→青 (非红)
8	1	1	1	→白

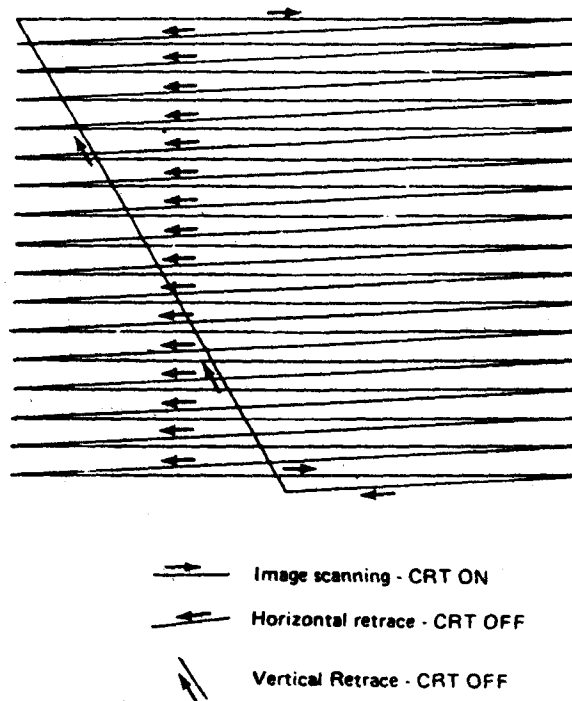


图 1-4 电子束的光栅扫描路径

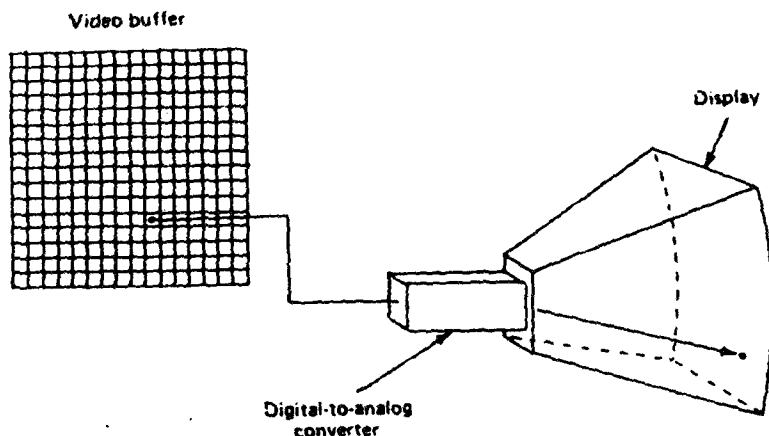


Figure 1-5. Memory-mapped, black-and-white display.

图 1-5 内存映象的黑白显示器

IBM 微型机上目前可用的所有图形硬件系统均为内存映象的光栅扫描类型。这些显示系统在第二章讨论。光栅扫描显示器的优点是廉价、彩色功能和它与用户或操作员的快速交互。它的主要缺点是显示器的物理结构所造成的：非水平、垂直或恰好 45° 角的直线的呈现出阶梯 (staircase) 效果。图 1-6 示出了象元结构 CRT 上的这种效果。

光栅扫描 CRT 在动画方面也受到限制。带来这个问题的有两个因素：每幅图象变化时，屏幕的所有象元都必须修改；为达到动画效果，屏幕必须每秒显示 20 幅图象。要处理这些问题，便给显示系统带来了很大的负担。只有技术上极为先进的硬件才能以足够快的速度进行这些更改，以获得满意的结果。

1.1.2 图象的物理性质

理想的计算机图形系统显示高清晰度的三维彩色图象，而且动画效果栩栩如生。但是，即使是大型机上最复杂、最昂贵的图形系统也尚未达到理想境界。微型机图形设备更是远远落在了后面。

计算机图象的目的是生成真实图象替身来传递视觉信息。替身通常以光的方式显示在 CRT 监视器上。这种图象的某些特性可科学地测量出来，至少可以客观地评价。可是，其它特性就没那么容易分析了，因为在感知图象中人的因素不易测量。例如，美学考虑可帮助判断某幅图象是否“看上去”比另一幅更好；而眼睛疲劳引起的头痛则可能会否定许多技术优点。在下列小节中，我们只考虑图象的可以科学评价的性质。图象感知的生理和美学因素将在本章稍后的小节中讨论。

1.1.2.1 亮度和对比度

亮度是一个表面单位面积反射光线的强度。人眼通过测试颜色值和亮度级别才能看见物体。增加物体的亮度也就增加了它的清晰度，但就其可视性而言，它与邻近区域的颜色差别或对比起物质的颜色或亮度的绝对数值显得更为重要。

一般情况下，普通人的视角清晰度估计可以维持约一分弧度。就是说，在 18 英寸远处看 CRT 显示器能看清千分之五英寸长的物体。一个有趣的现象是，8 英寸宽的显示器（如许多计算机系统中所用）每行可分辨的象元达 1600 之多。在流行的 IBM 微机图形系