

Windows NT 3.1 图形 编程技术

[美] Ben Ezzell 编著

傅祖芸 等译

电子工业出版社

(京) 新登字 055 号

内 容 提 要

本书是《Windows NT 3.1 编程技术》一书的姐妹篇。本书为程序开发人员提供了设计和开发 Windows NT 应用程序和图形应用程序的基本知识和技术。本书基于 Windows NT 的高级图形设备接口 (GDI), 系统地讲述了 Windows NT 的图形系统及功能, 彩色调色板, 图像数据格式, 图形打印, 图形增强、滤波及变换, 图像动画, 图形仿真等技术和编程方法。它为读者提供了丰富的高级图形编程工具。

本书还给出了大量的相应实用示例程序, 可以在任何 C 编辑器下编译, 其 C 源代码都装入一张 3 英寸软盘中。

Windows NT 3.1 图形编程技术

[美] Ben Ezzell 编著

傅祖芸 等译

责任编辑: 田野

*

电子工业出版社出版 (北京市万寿路)

电子工业出版社发行 各地新华书店经销

北京市顺义县天竺颖华印刷厂印刷

*

开本: 787×1092 毫米 1/16 开 印张: 22 彩色插页: 1 字数: 560 千字

1994 年 6 月第一版 1994 年 6 月第一次印刷

印数: 4000 册 定价: 43.00 元

ISBN7-5053-2212-5/TP·584

译者前言

Microsoft Windows 是当前最受欢迎的软件开发环境之一，许多著名的软件公司都在开发支持 Microsoft Windows 的软件。

《Windows NT 3.1 图形编程技术》一书是《Windows NT 3.1 编程技术》的姐妹篇。它们为程序开发人员提供了设计和开发 Windows NT 应用程序和图形应用程序的基本知识和技术，也是 Windows NT 程序开发人员所需的最新和最全面的必备资料。

《Windows NT 3.1 图形编程技术》是基于 Windows NT 的高级图形设备接口(GDI)，系统地讲述了 Windows NT 的图形系统及功能，彩色调色板，图像数据格式，图形打印，图像增强、滤波及变换，图像动画，图形仿真等技术和编程方法。它为读者提供了丰富的高级图形编程工具。

本书还给出大量的相应实用示例程序。这些例程可以用作为读者自己的 Windows NT 图形应用程序的样本，加以借鉴。本书的例程可以在任何 C 编辑器下编译，其 C 源代码都装入一张3英寸软盘中，供读者选购。

参加本书翻译工作的有：傅祖芸，段成华，许进华，汪爱媛，刘睿岚。傅祖芸和段成华两同志审校了本书的全部手稿。

由于译者水平所限，加上时间仓促，虽经认真审校，书中难免有错误和不妥之处，恳请读者批评指正。

译者
1993年11月

引 言

DOS 操作系统没有采用图形界面,其应用程序有许多不便之处:图形能力很难检测,驱动程序不可靠,总的说来,其结果是不可预测的。

然而,Windows 改进了以前 DOS 的不足之处。无论应用程序在干什么,它总是处于图形方式,图形能力已经检测过(如果 Windows NT 已经安装),驱动程序是稳定而可靠的,总之,虽不能说十分可靠,相对来说其结果至少是可以预测的。

但是,更重要的是,在 Windows 下,你的用户需要图形,而你的竞争者也正在为用户提供图形,因此你唯一实在的选择是加入激烈的竞争者行列或者是被人遗弃。

当然,你已经作出了正确的选择,对吧?否则你还有必要读这本书吗?

本书的内容

首先,如果读者想要学习设计 Windows NT 应用程序的基本知识,可先去查阅作者写的另一本有关 Windows NT 编程技术的入门性书《Windows NT 3.1 编程技术》。该书讲述了如何编写应用程序,Windows 如何利用事件消息来控制程序执行,以及在多任务环境下应用程序是如何工作的等基本知识。

然而,如果读者已经熟悉了这些基本知识,本书将从第一章开始,帮助你学习创建图形应用程序的过程,本书第一章回顾了图形设备的特点和能力,以及在 Windows 环境里它们的使用方法。

第二章进一步讨论了应用程序如何检测图形和其它系统设备的能力,有选择地讨论了应用程序如何把操作与系统能力相匹配(没有很大必要去关心系统设备能力限制的细节,但这些信息有时是有用的)。

第三章介绍彩色与彩色调色板以及演示,如何使对应用程序图形呈现的期望适应于不同视频硬件的可用资源。这一章还讨论了通常的调色板颜色条目和一些彩色转换(在第十二章将深入讨论调色板和彩色到灰度的转换)。

第四章讲述访问图形屏幕的高级技术,包括从屏幕上获取图像,并在应用程序中显示这些图像的方法。分专题介绍位图格式,压缩格式以及图像文件结构,重点介绍 Window NT 本身的图像格式。

第五章讨论了几种不同的图像格式,重点讨论目前流行的 PCX 格式,同时还介绍了 GIF、TIF 和 TGA 格式,以及 24 位(真彩色)图像转换。

有了获取、存储和显示图像的基础之后,第六章将介绍图像的增强、修正和图像数据滤波,包括边缘检测、变换、滤波处理,以及其它处理算法。

在显示和处理图像同时,我们希望能把图像打印出来,所以第七章便介绍作为一种设备描述表(Device Context)的打印机,并介绍用单色打印机打印灰度图像和用彩色打印机打印有限颜色表示的调色板彩色转换图像。

当然，图像本身仅仅是图形过程的一部分，在第八章里，图形光标和简单的动画将与过程定时机制结合起来，以便说明动画的原理和限制。

在动画受到限制的地方，交互式图像是另一种不同的解决方案。因为交互式图形的一个关键因素就是其在图像中识别区域的能力，所以在第九章里安排了两个例子，讲述在不规则区域内识别位置和坐标的能力。

第十章介绍商业用图形，除去简单的商业用图形之外，位图图像与传统的图形结合起来，即用线与条形图作为例子，来改善那些单调无味的图形呈现。并且从交互式商业用图形的角度进一步推荐了改善图形的方法。

在仿真领域中图形学作为一种工具变得日益重要，这是第十一章将要讨论的内容。讨论的题目包括设计考虑，使用彩色和定时器。在这一章里，一个“活的”森林被用来演示仿真一个小的但又可见的宇宙。这一章还讨论了加速计算密集型仿真方法。

最后，第十二章把题目转向彩色和彩色调色板，重点不仅仅放在目前十分流行的 SUGA 视频系统的内在关联性上，而且还讨论了24位“真彩色”视频系统，这将是明天的标准，我们将演示四种生成256种颜色的调色板：比例式调色板、加权式 RGB 调色板、色度饱和度强度式调色板和灰度式调色板。此外，还要讨论系统功能的一些限制。

附录 A 对扫描仪、视频摄像机和其他几种图形输入设备作了一些评价和建议。附录 B 说明了如何使用本书所带的软盘。

作者声明：我不是一个沿街叫卖的小贩（碰巧，本书正通过一个促销的电视节目销售），我很难保证，或者作任何建议说本书将使读者成为一个伟大的图形艺术家，或者成为一名练达的程序员，读者的才能、想象力和编程技巧不是我的能力所能影响的，然而，本书将会给读者提供一些有用的工具，帮助你成为一个更好的图形程序员，并满怀希望地祝你有更多的机会表现自己的才能，训练自己的艺术技能。除此之外，我只能祝愿你走运，并说一声“快乐”。

关于本书附带的软盘

本书最后带的软盘，包含有本书使用的所有示范程序，还包括一些示范程序所需的图像文件。附录 B 有该盘内的程序的详细说明。

目 录

引言

本书的内容

关于本书附带的软盘

第一章 图形系统..... (1)

图形硬件的特点..... (2)

入门知识 (2)

图形监视器与图形视频板..... (3)

存储器与视频分辨率..... (3)

图形协处理器..... (3)

用图形协处理器绘图..... (4)

Windows NT 图形 (映射) 方式 (4)

硬件定义的映射方式 (5)

Windows 映射方式 (6)

映射分辨率与显示分辨率..... (7)

设置映射方式..... (8)

作为虚拟屏幕的设备描述表 (11)

生命游戏程序 (11)

虚拟屏幕到物理显示设备的映射 (12)

程序注释 (13)

小结 (14)

TEMPLATE. I 程序清单 (14)

WINMODES 程序清单 (16)

LIFE 程序清单 (24)

第二章 系统图形能力识别 (32)

识别视频设备的能力 (32)

设备调色板 (彩色) 能力 (32)

GetDeviceCaps 函数 (34)

图形协处理器 (39)

其他视频图形支持 (39)

信息描述表与设备描述表 (39)

DC 演示程序..... (39)

小结 (44)

DC 程序清单..... (45)

第三章 彩色和彩色调色板 (59)

Windows 的调色板 (60)

定义彩色	(61)
标准调色板	(62)
COLOR1程序	(63)
利用抖动克服调色板的局限性	(64)
建立用户彩色	(65)
改变调色板颜色	(66)
灰度与彩色的转换	(71)
灰度调色板	(71)
真彩色灰度级	(72)
灰度级和方块效应	(73)
小结	(73)
COLOR1演示程序清单	(74)
COLOR2演示程序清单	(77)
COLOR3演示程序清单	(81)
第四章 图形实用程序	(89)
屏幕图像的捕获和显示	(89)
捕获屏幕图像信息	(90)
位图(.BMP)图像文件格式	(95)
写位图调色板	(100)
写位图数据	(102)
位图压缩格式	(103)
Windows NT 文件操作	(105)
CreateFile 函数	(105)
为命名的管道使用 CreateFile	(108)
WriteFile 函数	(108)
ReadFile 函数	(109)
覆盖文件操作	(110)
CloseHandle 函数	(110)
小结	(110)
CAPTURE 程序清单	(111)
第五章 几种图像格式	(121)
画笔的.PCX 格式	(121)
单色.PCX 图像解码	(124)
EGA/VGA 16色调色板信息解释	(124)
16色.PCX 图像解码	(124)
VIEWPCX 程序	(125)
VGA 256色调色板信息解释	(125)
256色.PCX 图像解码	(127)
每像素24位.PCX 文件解释	(130)

COMPUSERVE 的图形交换格式 (GIF)	(130)
标鉴图像文件格式 (TIFF)	(131)
TARGA (TGA) 格式	(131)
24位彩色图像的转换.....	(132)
彩色变换算法	(133)
TGA2VGA 程序	(134)
优化图像显示.....	(134)
小结.....	(134)
VIEWPCX 程序清单	(135)
第六章 图像增强	(150)
把图像转换成图.....	(150)
简化复杂信息	(151)
边缘检测算法	(155)
图像处理中的硬件.....	(158)
优化图像处理.....	(159)
实现处理算法; SHADES 程序	(160)
SHADES 程序中的选项	(160)
装载和显示位图图像.....	(160)
读取位图	(161)
Windows 3.1和 DOS 的图像处理	(168)
小结.....	(168)
SHADES 程序清单	(169)
第七章 打印图形	(191)
打印机操作.....	(191)
查询打印机	(192)
彩色与黑白打印机.....	(194)
从显示到打印机描述表的复制.....	(194)
InitPrint 函数, 版本1	(195)
打印机设置措施.....	(196)
禁止窗口	(197)
异常中止过程.....	(198)
打印.....	(199)
图像尺寸的测定.....	(200)
将彩色转换成灰色亮度.....	(201)
建立灰度级	(202)
不严格灰度级.....	(202)
彩色图像到灰度模式的映射.....	(202)
测定图像尺寸以避免花纹现象	(207)
控制打印机.....	(208)
灰度图像到彩色的映射.....	(210)

彩色打印	(210)
小结	(211)
GRAYIMG 程序清单	(212)
第八章 光标、位图和简单的图像动画	(226)
动画原理	(226)
定时技术	(226)
图像	(229)
控制技术	(229)
Animate1和 Animate2应用程序	(231)
小结	(231)
ANIMATE1程序清单	(232)
ANIMATE2程序清单	(241)
第九章 交互式图像	(253)
区域和界限识别	(253)
用颜色进行区域识别	(253)
使用醉汉走路算法	(256)
直接坐标搜索	(260)
醉汉走路与递归搜索	(260)
小结	(261)
MAPDEMO 程序清单	(261)
第十章 商业应用中的位图	(270)
一种简单的线图	(270)
在线图中加入位图	(273)
把位图与显示环境匹配	(274)
使用替换方法	(274)
条形图	(275)
饼图	(276)
加入与用户交互的内容	(277)
非商业应用中的图形	(278)
小结	(278)
BUSGRAPH 程序清单	(279)
PIEGRAPH 程序清单	(286)
第十一章 图形仿真	(292)
图形在仿真中的价值	(292)
森林宇宙	(293)
创建一个动态（有记忆）的宇宙	(293)
宇宙的规则和条件	(293)
解决有界的（有限的）问题	(294)
创建一个闭合的、无边界的宇宙	(294)

仿真中颜色的利用.....	(295)
进行森林仿真.....	(295)
初始化森林宇宙	(296)
可变的时间.....	(297)
仿真设计.....	(299)
扩展仿真.....	(300)
机械仿真.....	(301)
仿真理论实体.....	(301)
加速浮点运算.....	(302)
把虚拟屏幕用作存储器.....	(303)
小结.....	(303)
FOREST 程序清单	(304)
第十二章 SVGA 调色板和其他配色法	(316)
构造 Windows 调色板	(316)
调色板的局限性	(316)
比例调色板.....	(317)
RGB 调色板	(317)
HSI 调色板	(318)
灰度调色板.....	(319)
绕过缺省调色板.....	(319)
真彩色：16、24、32位彩色图像.....	(319)
小结.....	(321)
SVGA_WIN 程序清单	(321)
附录 A 机器视觉.....	(332)
扫描仪.....	(332)
视频摄像机.....	(332)
机动摄像机.....	(333)
视频设备附件.....	(333)
附录 B 关于附带的磁盘	(335)
插图部分.....	(339)

第一章 图形系统

《Windows NT 3.1图形编程技术》引入了一系列图形专题,并在 Windows NT(新技术)下,使用 Windows NT 本身的 API(应用程序接口)函数,以及 Borland 公司的32位 C/C++编译器 and 库扩展函数,讲解图形编程的原理。当然,由于 Windows NT 和其他先前的 Windows 版本一样,是一种图形环境,所以任何有关 Windows NT 编程的书籍或多或少都会涉及到图形问题。本书涉及的题目都是关于图形的,附带谈到文本显示、文件和目录操作以及其他非图形操作。

然而,如果读者希望得到有关 Windows NT 非图形操作的进一步的信息,或者要想了解 Windows NT 编程的基本信息,可去查阅本书的姐妹篇《Windows NT 3.1编程技术》(Ziff-Davis 出版社,1993年版),该书介绍了 Windows NT 图形设备接口(GDI)的基本特征,以及需要与 GDI 通讯的 API 函数调用。不过,《Windows NT 3.1编程技术》并不一定要与本书一起使用。

本书所用的例子包含在本书附带的软盘中,程序文件是以压缩的 .ZIP 格式提供的,当恢复时,将按程序或源文件所在章节的顺序建立相应的子目录。附录 B 和磁盘文件 FILELIST.TXT 提供了例子的清单。

有了以上的说明之后,我们立即转入正题:图形编程。我们先来看看 GDI 及其扩展,一般的图形与彩色操作。

常规的编程技术,即在 DOS 下编程,图形操作和图形应用都受到特定图形硬件的限制,或可以采取编程内部的防备措施去适应运行系统的硬件资源。然而,如何确定可用的硬件资源,总是一个主要的问题,即使有了进一步改进,如 Borland 公司的图形接口(BGI)已经排除了在适应不同硬件环境时所遇到的许多困难时,这个问题仍然存在。甚至于当系统的辨认是实际可行时,还有适应不同图形硬件分辨率的许多工作仍留给了程序员。

在 Windows 下,特别是在 Windows NT 下,GDI 采用了一种不同的方法,即提供一个环境壳,它沟通了应用程序的图形要求与正在运行的系统的图形功能。这样,利用 GDI,Windows 应用程序就能够从单色显示移至 VGA 显示或者移至真彩色显示,基本不需要应用程序去适应这种变化。不但如此,GDI 还能扩展对视频环境的支持,从而包括支持硬拷贝输出设备,这种特征在其他图形环境不具有的。在 Windows 下,应用程序在单独的虚拟环境下运行,独立于真实的系统硬件,是 Windows 把应用程序的虚拟显示映射到真实的物理显示。在这种方式下,运行在256种颜色的 SVGA(超级 VGA)上的显示图形可以被映射到16种颜色的标准 VGA 上,或映射到单色显示系统上。

当然,对 Windows 的支持也有一些限制。一些现成的设备,如3-D 显示器和彩色色散打印机,在使用中受到限制,现在 Windows 的驱动程序还不支持这类设备。随着更新的视频和硬拷贝设备的出现,开发商很快就会提供他们自己的驱动程序和改进接口,以增强 Windows 的能力。非常有意义的是,这些增强并不需要应用程序员操心,而由硬件生产厂家负责。

尽管 Windows 提供一个统一的虚拟图形环境,然而,真实的显示和硬拷贝的特性显然仍是取决于物理硬件设备的。在 Windows 把虚拟环境转换到一个物理环境之前,有关物理

设备的信息是至关重要的。一旦 Windows 得到这个信息,则它对应用程序也是有用的。

然而,有关图形设备特性的信息无论有多么详细,一般来说是没有什么用处的,除非是我们首次想知道我们将要用到什么样的图形硬件。因此,在我们检查可用的信息种类和了解如何访问这些数据之前,我们对硬件类型和它们的功能作一个简单的介绍。

图形硬件的特点

除了笔式绘图仪(以及其它同类设备)以外,所有图形设备都是点式的,即它们是基于像素或点的显示。实际上,视频监视器,与其它所有 CRT 显示器一样,都是采用光栅扫描(激光打印机也采用类似的机制),但计算机提供给显示用的数据却总是位图式的,它与真实的物理设备的机制是不对应的。因此,物理显示,即真实的视频监视器对我们来讲是次重要的。因为图形视频卡决定了视频分辨率。当然,有的 LCD 显示器(单色的或彩色的)是真正的数字设备,即使如此,它们还是取决于与图形视频卡类似的硬件设备。图形卡仍然比物理屏幕更加重要。

入门知识

几年以前,人们还对微型计算机感到新奇的时候,IBM 公司推出了 CGA(彩色图形适配器)视频适配器。这种最初的视频适配器能支持图形(最大分辨率为 640×200 个像素),但它的分辨率和彩色是有限的。而先前的显示设备仅限于支持显示文本的 CRT;或在许多场合下,仅限于支持行式打印机,因此即使是这些原始的图形显示设备也被广泛地称为重大的成就,直到后来许多程序员发现这些 CGA 显示设备的真正限制为止。

更先进的视频卡相继推出:首先推出具有 640×350 个像素分辨率的 EGA(增强型图形适配器),其后又推出了具有 640×480 个像素分辨率的 VGA(视频图形阵列),以及具有 1024×768 个像素的 8514/A。其中 EGA 卡仍能在新机器上见到,但用得已不多了。8514/A 并不太受欢迎,这主要是因为它的价格和兼容性问题。今天,VGA,特别是 SVGA,已经成为最流行的标准。VGA/SVGA 卡可以有多种配置和功能。

表 1.1 给出了四种主要的视频卡的主要特征。这些信息摘自 DC(设备描述表)程序,该程序在《Windows NT 3.1 编程技术》一书中介绍过,本书第二章也有介绍。每一种视频卡/监视器所报道的准确数据,随设备、配置和方式设置不同而有所变化。

表 1.1 典型的视频卡分辨率

功能	测量单位	CGA	EGA	VGA	SVGA
HORZSIZE	毫米	240	240	240	280
VERTSIZE	毫米	180	175	180	210
HORZREZ	像素	640	640	640	1024
VERTREZ	像素	200	350	480	768
ASPECTX	...	5	38	36	14
ASPECTZ	...	12	48	36	14
纵模比	xy 比值	0.416	0.791	1.000	1.000

续表

功能	测量单位	CGA	EGA	VGA	SVGA
ASPECTXY	对角线	13	61	51	19
LOGPIXELSX	像素/英寸	96	96	96	120
LOGPIXELSY	像素/英寸	48	72	96	120

* 所列数值是计算的结果, 不是所报告的比值

图形监视器与图形视频板

图形监视器基本上是独立于驱动它们的视频板。即使是很老式的视频监视器也能与现在的图形板兼容, 并能在高分辨率的视频方式下显示图形。

例如, 几年前我买了一台多频监视器, 原是为配置有 EGA 视频卡的机器用的, 后来又用于带有 VGA 和 SVGA 视频卡的机器。虽然, 当初这台监视器是作为 640×350 彩色监视器设计的, 但最终用的配置能达到 1024×786 个像素、256 种颜色的分辨率 (需对场同步作些小的调整, 以适应最高分辨率方式)。其它老式的监视器也是如此, 也需作些同步调整, 否则不能匹配所要求的同步速度。

存储器与视频分辨率

对于高分辨率显示, 至关重要的是有足够大的存储器, 不是主板上的存储器, 而是指视频卡上的存储器。对于 Windows 操作, 主板上至少应有 4MB 存储器, 但这些存储器并不是用于视频显示的。SVGA 图形卡上的视频存储器比较理想的情况是应有 1MB 的容量, 以便能支持 1024×768 、256 种颜色的视频显示。有不少 SVGA 卡只配置了 512KB 的 RAM。有的卡在销售时只配置了 256KB 的 RAM, 这是不适当的。但可以升级到 512KB 的容量, 以支持 640×480 、256 种颜色的视频显示。

注: 1024×768 、256 种颜色分辨率要求完整的 1MB 的 RAM。

将来, 人们能看到图形视频卡有更高的分辨率, 特别是新的 24 位彩色图形板, 能提供真彩色图形 (在视觉极限范围内)。每块真彩色板有大容量存储器和某种图形协处理器。除此之外, 它还提供自己的 Windows 驱动程序, 这种考虑令图形程序员更加乐观。

图形协处理器

图形协处理器是 1992 年最时尚的高技术话题, 市场红火, 必定会主导今后的潮流。在我们讨论图形协处理器是什么, 能干什么之前, 让我们首先来了解一下如果没有协处理器, 图形操作是如何工作的。

在传统的图形显示中, 为了在屏幕上画一个圆, 应用程序必须计算显示一个圆的各点的位置。实际上, 除非采用汇编语言, 否则相对来讲, 这是件较容易的编程工作。通常, 它只需要一个函数调用, 如 C++ 的 circle 函数, 只需输入圆心位置和半径的参数即可 (Windows 的函数稍有不同, 后面将讲到)。虽然操作很容易完成, 然而, 执行该操作却需要大量的运算, 占用不少 CPU 时间。所有图形操作都是如此, 即使是象在屏幕上画一条线也是如此。类似地, 图形文本 (采用笔划字体) 比常规的 DOS 文本需要更多的时间来显示, DOS 文本采用位图式字体。

即使图形显示比基于字符的显示速度慢,但在许多实际应用场合,其显示速度仍是可以接受的。对于这种说法,程序员们会觉得这是一种限制,而不是一个可以接受的标准。许多程序员,就象赛马手一样,特别珍惜节省系统任何一点可能利用的时间,即便是一些片断时间。这些片断时间对用户来讲实际是看不见的,只能利用很高级的剖析测试工具才能检测到。能增加额外的几微秒的系统速度,正是图形协处理器发挥作用的地方。

令人高兴的是,新推出的图形板降价很快,能与先前推出的较简单的图形板竞争。但是为了获得一点点利益而购买简单的图形板,这是件不明智的事,因为对多数程序员来讲,价格差并不是首要考虑的事情。多花几块钱得到一块1MB的视频卡替代256KB的视频卡是很合算的投资。或者,如果当遇到繁重的图形任务,并且要求好的图形显示效果和性能,例如要编辑照片、艺术作品,或者完成工业级的图形设计,那么,无论如何都应多花点钱购置一块真彩色图形板。它们应不只是支持256种颜色的调色板,它们还应为一幅1024×768或更大的图像的每一个像素提供24位彩色定义。

用图形协处理器绘图

目前,尚没有评价图形协处理器性能的明确标准,图形协处理器这个术语被用于功能各异的设备中。然而,一般来说,图形协处理器是视频图形板上的专门处理芯片,它完成各种绘图操作的运算任务。

当安装了适当的图形协处理器后,应用程序并不需要去计算绘制一个圆的各个点,而是发送一条指令给视频板,并送入圆心坐标和半径,请求作一个相应的圆(类似先前提到的用调用C++的circle函数实现)。作为响应,图形协处理器用它自己专门的硬件算法来计算所需的点。硬件操作因其固有的优点,比软件操作快得多。图形协处理器支持的操作也非常宽广,但一般包括以下五种:平面、线、多边形、曲线和文本操作。在每一种操作中,所支持的功能变化也较大。

使用Windows时,应用程序并不需要知道在一特定系统中的图形协处理器的情况。在安装时,Windows已识别了可用的硬件,就不需要去利用系统能够提供的最高级硬件能力。虽然应用程序并不要求去直接识别视频设备能力(或硬拷贝设备能力),然而知道什么硬件支持可以利用,仍然是有用的。如何识别支持的特性将在第五章中讨论。下一节将讨论更直接的题目:了解Windows使用的图形方式。

Windows NT 图形(映射)方式

前面已强调了,使用Windows,程序员可以从识别系统提供的图形硬件能力的任务中解脱出来。现在我们进入有关Windows NT图形方式的章节。

首先,我们讨论的是两个十分不同的题目。请记住,在Windows NT(或Windows 3.x)下,当应用程序向屏幕写时,它实际上是在写一个虚拟的显示屏幕,而这完全独立于物理硬件。

其次,决定物理显示的特点(即物理硬件)正是Windows的责任。Windows映射虚拟显示的内容到物理显示,或至少对应用程序的窗口是可见的。虽然,在需要时,应用程序能够查询物理设备能力,但应用程序并不是一定需要关心物理设备的限制。然而,虽然应用程序

在写虚拟显示，但是 Windows 的映射方式已定义了虚拟显示的大小及定义了虚拟显示的内容是如何转换到物理显示设备上的。更重要的是，给出设备的独立性后，应用程序现在就可以自由选择或定义最适合任务的映射方式，而不是从物理方式和限制中去选择。

硬件定义的映射方式

映射方式这个词似乎有点陌生的感觉，其实，即使是早期的计算机视频系统就已经采用了。先前，只有一种，有时或者有几种映射方式被采用，它们由硬件（图形卡）决定，而不是由视频系统决定。

常规的 DOS 文件显示是最早的，也是我们最熟悉的映射方式，它映射一个基于 ROM 的字符集（字符像素阵列）到视频图形系统。请注意，即使是在很早的系统中也用这种方式，这些早期的系统并不支持真正的图形方式，但不管系统的限制如何，显示本身在本质上仍然是图形的。所以，在 DOS 文本显示方式中，映射方式是硬件决定的。以 CGA 系统为例， 8×9 像素字符阵列被映射到 640×200 视频图形中。在大多数 EGA 系统上， 8×14 字符阵列被映射到 640×350 视频显示中。对大多数 VGA 系统， 8×19 字符阵列被映射到 640×480 视频显示中。另外，对于 EGA 系统，用 43 行文本方式；VGA 用 50 行文本方式；而图形系统改变映射方式和字符集，从用于一个 25 行显示中恢复映射 8×9 的字符。

注意，SVGA 视频系统能够支持 1024×768 个像素显示。在某种场合下，能够支持 1280×1024 个像素显示，但这些卡仍继续支持所有早期的分辨率，包括 800×600 ， 640×480 ， 640×320 ， 640×200 以及 320×200 。

对于 Windows，不同的分辨率在 Windows 设置程序装入后可以通过改变视频驱动程序来建立，通常，较低的分辨率是不提供的。如果用户不能确信视频系统是如何配置的，Windows NT 设置程序〔放在主组（Main group）里，缺省值〕将会显示出当前的视频选择，如图 1.1 中的显示区所示。在设置阶段，选择 Options（选项）菜单，将显示另外一个对话框，允许选择几种系统设置。对于视频方式选项，所选择的对话将给出一个下拉式菜单，其内列出所有当前安装的驱动程序和映射方式。还有一个选择，以便从外部磁盘中装入新的驱动程序。

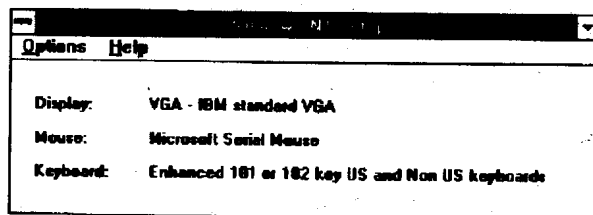


图 1.1 Windows NT 的设置对话框

许多 SVGA 卡要求使用自己的驱动程序。这些驱动程序随卡提供，它们常会提供为安装 Windows 驱动程序的工具和一些其它实用程序。这时就应使用这个安装程序。倘若使用的是标准视频卡，Windows 设置程序将允许在不同视频方式选择中切换。

请记住，在某个时刻，只能使用某一种硬件设备方式，即使是几个 Windows 应用程序同时使用不同的 Windows 图形方式时也是如此。

Windows 映射方式

在 Windows 下, 所支持的映射方式是很不一样的, 重要的是, 这些方式独立于硬件给予的限制。Windows 支持八种映射方式, 每一种方式对屏幕(窗口)坐标位置提供一种不同的测量单位, 如表1.2所示。

表1.2 Windows 映射方式

映射方式	方式 ID	逻辑单位	屏幕原点缺省值
MM_TEXT	1	像素	左上
MM_LOMETRIC	2	0.1mm	左下
MM_HIMETRIC	3	0.01mm	左下
MM_LOENGLISH	4	0.01英寸	左下
MM_HIENGLISH	5	0.001英寸	左下
MM_TWIPS	6	1/1440英寸	左下
MM_ISOTROPIC	7	可变($x=y$)	可变
MM_ANISOTROPIC	8	可变($x<>y$)	可变

MM_TEXT 方式(缺省值)本质上是与 DOS 的文本映射方式等同的。差别在于, DOS 是映射字符位置, 而 MM_TEXT 方式允许应用程序按照设备像素来工作。如同 DOS 文本方式, 屏幕(窗口)原点缺省值为左上角, x 和 y 坐标分别向右和向下增加, 亦正如在常规的 DOS 或 DOS 图形中的情形一样。Windows MM_TEXT 方式在本质上是与 DOS 一样应用于同一个目的: 用于文本显示(文档、表格、列表和报告等)。

然而, 从 DOS 的标准来看, Windows 的其它映射方式截然不同, 这是因为其缺省的屏幕原点位于窗口的左下角, 坐标值分别朝上和向右方向增加(本质上是笛卡尔坐标系)。在 MM_ISOTROPIC(各向同性)和 MM_ANISOTROPIC(各向异性)方式中, 原点可以位于有效窗口里面和外面, 但坐标系仍是笛卡尔坐标系。

在两个英制和两个米制方式中, 坐标位置是以英寸(in)或毫米(mm)定义的, 而不是以像素定义的。这对于某些应用来说是很方便的, 这些应用要求显示对应于实际单位的图形(如图案设计程序)。实际上, 即使在像素方式中, 如 MM_TEXT, 各种制式转换能用于同样的目的。但用英制或米制方式时, 实际单位到像素显示的转换是留给 Windows 来完成的, 而不必去担心设备的分辨率。

MM_TWIPS 是另外一种方便的方式, 这种方式用于打印设置程序。TWIPS 可以理解为“一个点的二十分之一”, 一个点就是打印机打印尺寸的度量, 72个点等于1英寸。所以, 一个 TWIPS 单位就等于1/1440英寸, 一个10点的打印面能用200个逻辑单位(TWIPS)的分辨率绘出, 而不用担心真实的显示设备能否有同样的分辨率(大多数情况下都不具有)。图1.2对几种映射方式作了比较。

最后, 各向同性的映射方式和各向异性的映射方式并不是为特定场合设计的, 这两种方式提供可变的逻辑单位和可变的坐标原点。这样, 当要想绘出一正弦波时, 原点可以放在窗口中间偏右的位置上, 这样便允许绘制的点可以在纵向取正和负值, 而在水平方向仅允许取正值。

在各向同性方式中, x 和 y 轴的单位是相等的。当要求维持一幅图的相对宽幅比时, 这点尤其重要。另一方面, 各向异性方式允许设定不同的 x 和 y 轴的坐标刻度。例如, 这种方式可用于绘制 x 轴以毫秒为单位, y 轴以伏特为单位的波形。也可以时间轴定义为 1000 个单位 ($=1$ 秒), 而纵轴定义为 ± 5000 个单位 ($=\pm 5$ 伏特)。这样, 就能画出电压对时间的波形, 而不再转换成对应于真实显示分辨率的单位。

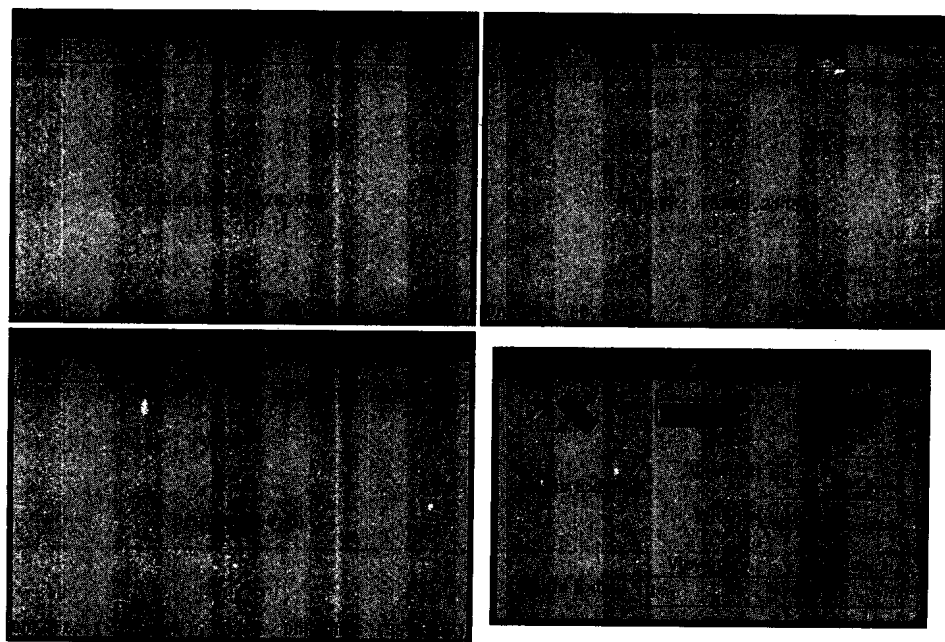


图1.2 文本、TWIPS 和各向同性的坐标映射

映射分辨率与显示分辨率

有了比显示分辨率高得多的映射方式, 这是很好的, 但也带来两个问题: (1) 具有这样高的分辨率的目的是什么? (2) 用这些分辨率如何显示映射图像?

首先回答第二个问题。如果我的回答听起来就象在回避问题, 那就只有请你谅解了。简单的事实是, Windows 从高分辨率描述表 (设备描述表或 DC) 中映射图像点到真实显示设备或打印机的分辨率上。它与所使用的精确算法是不相关的, 却是随使用的图像数据的类型 (线、面、字体或者位图) 而改变。

对于第一个问题, 有许多原因允许高分辨映射方式存在。主要原因在讨论 TWIPS 和讨论用各向异性方式显示电压时间波形时已经提到了。当设备描述表分辨率被映射到要绘制的数据时, 并不需要完成为匹配特定的视频、打印机或者绘图仪的能力而进行的转换。相反, 数据被绘到一个最适当的地方, 然后其结果被映射到真实的设备上, 其中所有需要的转换都由 Windows 完成。

采用高分辨率映射的第二个原因就是, 各种设备分辨率很不一致, 视频设备的分辨率与其他设备的输出分辨率相比, 通常是较低的。表 1.3 结合具有代表性的 Windows 映射分辨率, 比较一些视频和硬拷贝设备的典型分辨率。