

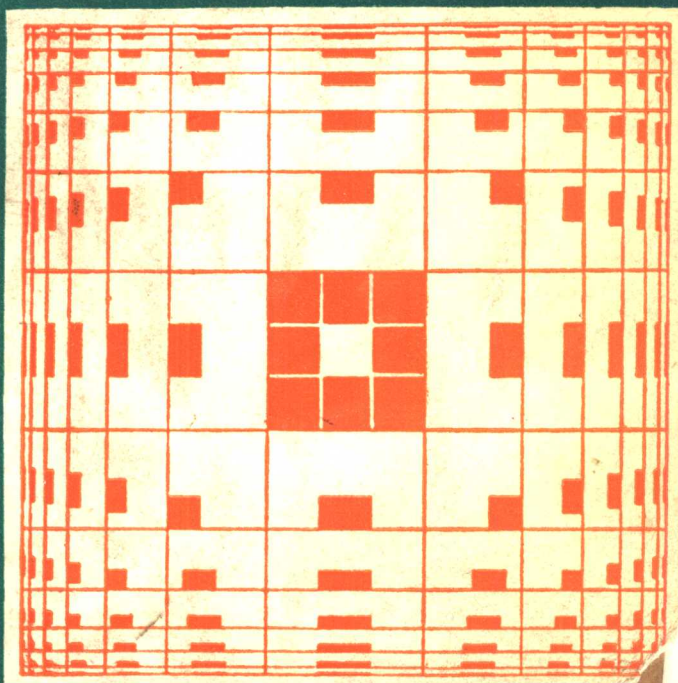
高等学校教学参考书

APPLE-II PASCAL

微型计算机辅助制图

胡瑞安 周济 常明 编

胡瑞安 主编



高等教育出版社

等学校教学参考书

APPLE-II PASCAL 微 型 计 算 机 辅 助 制 图

胡瑞安 周济 常明 编

胡瑞安 主编

高等教育出版社

内 容 提 要

计算机辅助制图是微机应用的重要领域之一。近年来,不少大专院校、中专校在教学中都增加了计算机图学的选修课,特别是微机应用的普及、深入,所用语言已不限于 BASIC、FORTRAN,不少单位正在使用 PASCAL 等语言来编制程序。

PASCAL 语言是目前世界上广泛流行的程序设计语言之一,它除了适用于科学计算外,还更适合于教学、书写系统软件与应用软件。

本书是使用 PASCAL 语言讲授计算机制图的教科书,又是学习 CAD 的重要基础。全书共分十章,包括二、三维图形处理的基本技术,几何变换,机械图样的绘制,自由曲线与曲面, CUBICS 程序及其说明,图形输出与数据结构分析, Apple-II PASCAL 基本操作,隐藏线消除,等等。本书可结合国内广泛使用的 Apple-II 微机进行学习,书中由简到繁地编有一系列 PASCAL 程序,书末还有美国斯坦福大学研制的用以绘制自由曲线的 CUBICS 程序供参考。

本书可供大专院校学生、工程技术人员学习 PASCAL 语言和计算机图学使用。

高等学校教学参考书

APPLE-II PASCAL

微型计算机辅助制图

胡瑞安 周济 常明 编

胡瑞安 主编

高等教育出版社出版

新华书店北京发行所发行

河北省香河县印刷厂印装

开本 787×1092 1/16 印张 16.5 字数 370 000

1988年8月第1版 1988年8月第1次印刷

印数 0001— 3,310

ISBN 7-04-000927-7/TH·56

定价 4.80 元

前 言

微型计算机在我国正在迅速普及,目前,不仅在高校、设计研究单位,而且在工厂以及中小学亦开始配备计算机。其应用范围也在不断扩大,由科学计算扩大到设计与制图等方面。所用的语言也不限于 BASIC,不少用户正在使用 FORTRAN、PASCAL 等语言来编制程序。

计算机制图是计算机应用的重要领域之一,图形作为输出的形式,具有生动、直观的优点,特别适用绘制各种实验数据、曲线与曲面、工程图样等,用来进行辅助设计与辅助教学。随着微机的普及,人们对计算机制图的兴趣也在逐年增长。近年来,不少大专院校及中专校,在工程制图教学中都增加了计算机图学的有关章节作为必修内容,有的院校还增设了选修课作为机械制图的一门后续课程。随着计算机图学教学与科研的发展,80年代初,有些高等学校已开始招收这方面的研究生。

中国工程图学会计算机图学专业委员会组织编写了一批有关教材,以满足选修课及研究生教学的需要。本书是用 PASCAL 语言编写,以供读者选用。

本书初稿于 1984 年作为华中工学院校内研究生与本科大学生选修课教学中试用的,后来根据试用情况进行了修改、补充或重写。并在有关 CAD/CAM 学习班上使用,配合使用美国斯坦福大学 Wilde 教授带来的一套用于 Apple-II 机的应用教学软件,取得了较好的效果。本书附录中所附的 CUBICS 程序,就是其中的一个主要组成部份。

本书共十章,其中,第一、二、五、七、九章由胡瑞安编写,第三章由周济编写,第四、六、八、十章、附录一、二和附录三的中文注释部分由常明编写。本书由胡瑞安主编。为了便于初学者参考,书中附有一部分程序,这些程序中不少地方是值得进一步改进与完善的,因此仅供读者参考,具有一定水平的读者则可以跳过这些内容进行学习。

在编写本书的过程中,承蒙华南工学院曾大民教授审阅,并提出了许多宝贵的意见,使本书质量有了提高,我们对此表示感谢。华中工学院机械工程一系 CAGD 研究生宋启付、张远红、张文祖、宦健、肖临清等积极为本书提供素材、验证算法、上机调试、并绘制了一部分插图,使本书得以如期完成。本书插图的描图工作,绝大多数都由黄谦同志完成的。书末附录中的 CUBICS 程序,是美国斯坦福大学 D.Wilde 教授设计并提供的,在此一并表示感谢。

限于编者水平,书中错误与不妥之处,尚希读者批评指正。

编者

1987. 9 于华中工学院

目 录

第一章 绪论	1	第六章 CUBICS 程序及其说明	113
§ 1-1 概述	1	§ 6-1 CUBICS 程序的结构	113
§ 1-2 图形处理与形状处理的研究内容	2	§ 6-2 CUBICS 程序的数学方法	115
§ 1-3 图形显示设备及其特征	3	§ 6-3 CUBICS 程序的数据结构	118
§ 1-4 计算机图形软件标准化的基本概 念与内容	5	§ 6-4 CUBICS 程序中几个有特色的处 理方法	127
第二章 图形处理的基本技术	10	第七章 自由曲面	133
§ 2-1 概述	10	§ 7-1 自由曲面的描述	133
§ 2-2 直线的生成算法	10	§ 7-2 双线性曲面	134
§ 2-3 圆的生成方法	13	§ 7-3 Coons 双线性曲面	135
§ 2-4 二维剪取	15	§ 7-4 Ferguson 双三次曲面	138
§ 2-5 非圆曲线的生成	18	§ 7-5 Coons 双三次曲面	139
§ 2-6 二维仿射变换及其性质	22	§ 7-6 Coons 双三次曲面的形状控制	141
§ 2-7 二维射影变换及其性质	25	§ 7-7 Bézier 曲面	144
§ 2-8 视区与窗口	29	§ 7-8 Bézier 曲线与曲面的连接	148
§ 2-9 例图及其变换	31	§ 7-9 B-Spline 曲面	149
第三章 几何变换	33	第八章 图形输出与数据结构分析	153
§ 3-1 几何变换概述	33	§ 8-1 屏幕显示与打印机输出图形	153
§ 3-2 二维几何变换与齐次坐标系	33	§ 8-2 SR-6602 型绘图机接口技术	159
§ 3-3 三维图形的基本变换及组合变换	39	§ 8-3 SR-6602 型绘图机主要指令及其 调用方法	161
§ 3-4 三维图形的正投影变换	44	§ 8-4 图形输出与程序设计方法	164
§ 3-5 三维形体的透视投影变换	54	§ 8-5 规则子程序包与内部子程序包的 设计方法	171
第四章 机械图样的绘制基础	65	第九章 隐藏线的消除	176
§ 4-1 概述	65	§ 9-1 基本概念	176
§ 4-2 PASCAL 图形程序库	68	§ 9-2 多面体及其描述	176
§ 4-3 基本几何图形程序设计	68	§ 9-3 消隐的基本方法	180
§ 4-4 剖面线程序设计	76	§ 9-4 窗口再分算法	185
§ 4-5 尺寸标注及公差配合程序设计	77	§ 9-5 扫描线算法	187
§ 4-6 标注表面粗糙度的程序设计	82	§ 9-6 将多面体分割为凸多面体集合的 算法	190
第五章 自由曲线	83	第十章 APPLE-II PASCAL 基本操 作及 CUBICS 等软件的使用说明	192
§ 5-1 概述	83	§ 10-1 APPLE-II PASCAL 基本操作	192
§ 5-2 Hermite 插值曲线	84	§ 10-2 PASCAL 程序的编译	197
§ 5-3 Bézier 曲线	92		
§ 5-4 均匀二次 B-Spline 曲线	99		
§ 5-5 均匀三次 B-Spline 曲线	101		
§ 5-6 B-Spline 函数及其性质	107		

§ 10-3	PASCAL 程序的链接	199
§ 10-4	创建新系统库	201
§ 10-5	圆及其他参数曲线	203
§ 10-6	Hermite 插值曲线	205
§ 10-7	Bézier 曲线	207
§ 10-8	B-Spline 曲线	209

附录一	IBM-PC 微机绘图包与 APPLE-II 绘图包	212
附录二	PASCAL 语法简表	215
附录三	CUBICS 源程序	217
参考文献		255

第一章 绪 论

§ 1-1 概 述

微型计算机已在我国广泛使用。近年来,在工程设计中,不仅数值计算,而且体现设计者在设计制图方面的创造性工作中,也正在以各种形式与不同程度上使用着计算机。在四化建设中,设计制图的计算机化已经成为必然的趋势。

计算机图学是计算机学科领域中的一个分支,在人类感受的信息中,70%左右是图象信息,如何用计算机处理并生成各种图形,始终是人们注目的课题之一。对于人来说,计算机图形显示给人以生动、直观的感觉,因而也是最容易被人所接受与理解的。计算机图形能形象地、动态地反映客观世界的运动、变化与联系,有力地弥补数值输出的不足,因此,不仅科研、工程设计、生产制造中,而且艺术设计以及教育训练中,都在广泛使用。很显然,人们从图形中接受信息的速度确实比扫视一大堆数字表格要快得多,因此,随着硬件价格的降低,其应用领域也在不断扩大。

图形不同于数值,是一种非数值信息。因此,对于计算机来说,图形处理不仅方法上,而且在外观上、动态上都不同于其他方式的处理。为了在设计与科研中使用计算机图形,必须掌握图形处理与形状处理的有关知识。

为了用计算机产生与显示图形,要有一组几何数据(例如,立体各顶点的坐标值),但这还不够,还必须要有表示各顶点的连接关系。上述两类信息(数据),构成了定义该形状(或图形)的数学模型。

图形处理的内容是:当给出形体的数字模型时,如何用计算机来定义、存储、操作,从而生成所需的图形信息,并在图形显示终端或自动绘图机上输出图形。通常称在图形终端上得到的图形处理称为图形显示,而把在绘图机上得到的图形处理称为自动绘图。二者统称为计算机制图。

由于计算机的内存是一维的,而要求描述的形状模型则是三维的,因而用计算机处理三维形状,具有一定的困难。将形状的数学模型在计算机内进行定义并进行的各种处理,称为形状处理或几何构型。

从60年代初到70年代中期,图形处理的研究有了一定的成绩从70年代中期到目前为止,伴随着CAD/CAM的发展,关于形状处理或几何构型的研究又有了较大的发展。这期间出现了不少图形系统和几何构型系统。目前正在迎接第三个研究高潮,这就是与图形与形状输入技术密切相关的、关于计算机视觉及其应用的研究。

§ 1-2 图形处理与形状处理的研究内容

作为一本教材,本书不可能对计算机辅助制图所涉及的所有问题作广泛的叙述,但在学习后述各章之前,有必要对它的两个主要组成部分——图形处理和形状处理的内容有一个概略的了解。现将这两部分的研究内容概述如下:

一、图形处理

为了在显示器屏幕上产生图形输出,需要将有关数据输入计算机,这些数据,一般都需要经过某种变换处理,才能把用以描述图形的坐标系(称为自然坐标系或世界坐标系)转换为用屏幕坐标系表示的数据。因此,在图形处理中,首要的问题是各种变换问题。例如,图形的放大、缩小、平移、旋转以及各种投影变换等。

在上述变换中,可以概括归纳为三种,即线性变换、仿射变换和射影变换。其中,除线性变换外,其他两种变换较复杂,而且还有各种复合变换。为了使问题简化,引入了齐次坐标的概念。齐次坐标的引入,不仅能有效地处理无穷远几何元素,而且可使所有变换处理用统一的形式来进行。因此,齐次坐标技术是图形处理的重要技术之一。

在图形处理中,如何选取可见信息,以便在显示图形时在屏幕上给人们提供一个灵活的观察窗口,显示各种投影图、局部放大图等。通常,可以设置一个矩形表示图形的领域,称为视区。当视区中的图形经过放大或其他变换之后,一部分图形不在此矩形域内时,则在屏幕上就不显示这一部分图形。处理图形元素并将其可见部分显示出来的技术,称为剪取,它也是图形处理的基本技术之一。

在计算机制图发展的早期,在表示三维图形时,人们采用线框模型,把所有棱线和轮廓线全部显示或由绘图机画出,这样的图不但很难看懂,而且存在着二义性。因此,有必要将那些看不见的线和表面,在处理时将其消除。这就是隐藏线(面)的消除。消隐是计算机图形处理中的难点之一,当形状的复杂性增加时,需要大量的运算时间。目前为止已对消隐问题进行了大量的研究,提出了各种算法,有的算法已由硬件来实现。但消隐算法仍是人们感兴趣的研究题目之一。

二、形状处理

形状处理主要研究三维形状信息的描述与处理的有关问题。

在计算机内部构造三维形体的几何模型涉及两方面的问题:一是形状的定义与描述问题,另一是数据结构的问题。这种几何模型必须尽量保持“形状”所具有的性质,如:空间的唯一性、形状的有界性、连续性,以及坐标独立性(即坐标变换后形状的不变性)等等。

如何描述构成形状的基本几何要素——表面的问题。一般工程中,用到的绝大多数是平面和二次曲面,它们可以用数字方程描述,这类表面所组成的形状称为第一类形状。另一类形状如汽车车身与飞机外形等,由自由曲面所组成,这类形状称为第二类形状。如何对这类自由曲面进行数字描述的问题,早在数控发展的早期就提出来了,此后一直被作为形状处理的中心问题之一而进行研究。60年代初期,航空工业中广泛应用着 Ferguson(法格森)曲面,1964~

1967年又发展了Coons(康斯)曲面,它们都是基于Hermite(赫米特)插值的构造自由曲面的一般方法,在计算机辅助设计与制造中起着重要促进作用。这类曲面的设计与形状控制中,由于需要非常熟悉各种矢量对形状的影响,才能更好地对形状进行控制。为了改善其控制性,60年代后期发展了Bernstein-Bézier(伯恩斯坦-贝齐尔)曲面,这是一种基于Bernstein逼近的构造自由形状的方法,其控制性比上述方法有较大的改善,但连接性有待改进,是一种整体逼近的方案。Bézier曲面已成功地应用于汽车工业。70年代初Gordon Reisenfeld(戈登-里森弗尔德)使用B-Spline(B-样条)基函数发展了B-Spline逼近的理论与构造自由形状的方法,它能保存样条曲线的优良的连接性,又具有Bézier曲线的控制性,因而被广泛使用。

鉴于由各种平面和曲面来构成机械零件时,其拓扑连接关系的复杂性,其中包括各种拓扑信息如何在一维的计算机存储器中的表示的问题,1974年提出了Baumgart(鲍姆加特)翼边数据结构(即WED结构),可使形状的构造、元素的增删等处理以统一的方式进行。这种数据结构不仅适用于多面体,也适用于曲面体的形状处理。

形状处理中还广泛地使用着布尔运算的方法,当形状采用实体模型时,通过布尔运算可以获得各种复杂的形体。尽管从实用的角度还存在某些问题,但理论上各种形状的处理与构型都是可以实现的,具有较高的应用价值。

以上内容,是图形处理和形状处理的大致内容,在本书的后续各章将具体加以介绍。本书将侧重讲述图形处理部分。上机实验几乎包含以上所述的大部分内容,可以作为读者进一步学习的入门向导。

§ 1-3 图形显示设备及其特征

计算机图形输出通常有两类工作方式,其一是屏幕输出图形,另一是绘图仪输出图形。前者使用比较普遍,图形在阴极射线管(CRT)的屏幕面上显示,显示速度快,修改后又可以在屏幕上显示出来,因此使用起来很方便,大多数微机用户都乐于采用这种方式。但这种方式只将图形显示在屏幕上,为了得到硬拷贝即纸面上的图形,有必要采用绘图仪来输出图形或其他方法得到硬拷贝的图形。微机一般使用小型的绘图仪。

屏幕显示就工作原理来分,有随机扫描和光栅扫描两种。前者又分两种,即刷新型随机扫描图形显示和存储管型随机扫描图形显示。图1-1为随机扫描图形显示的示意图。

1. 存储管型(随机扫描图形显示设备)这是一种速度比较低、价格便宜的显示设备,在其上显示图形后,画面难以修改。在设计中,若要对图形作局部修改,则必须将图形全部清掉,然后重新显示。它的显示速度,比刷新型显示设备低,适用于处理比较简单的图形。

2. 刷新型随机扫描图形显示设备 这种设备与Sutherland(萨瑟林)最早用于SKETCH PAD系统的设备类似,显示速度较高,并可对图形进行局部修改。交互性图形处理功能也较强,适于处理三维图形显示。但主要问题是当图形复杂时,画面上会产生闪烁。近年来几经改进,出现了一些水平较高的刷新型显示设备,由于系统复杂,价格比较贵。

上述图形显示设备都是以电子束在荧光屏上沿图形的轨迹移动而产生图形的,故称为随

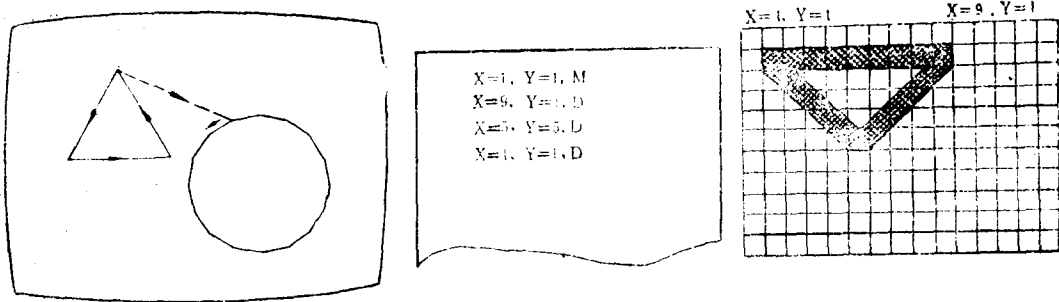


图 1-1 随机扫描图形显示

机扫描显示。

3. 光栅扫描图形显示设备 图 1-2 为其示意图,这种显示方式由于结构简单,价格便宜,故应用日益广泛。图 a) 为行扫描的示意图。

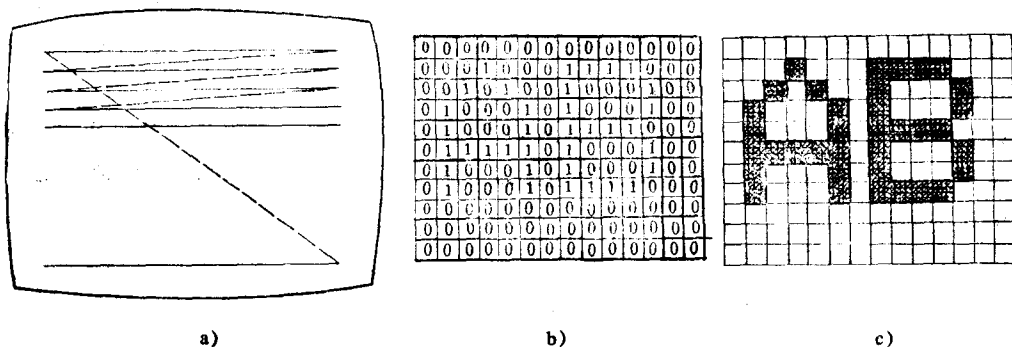


图 1-2 光栅扫描图形显示

光栅扫描显示设备又称为刷新型光栅扫描图形显示器,它具有多种描述图形的功能,既可描绘线框图形,又能显示彩色浓淡图象。它主要由帧缓冲器、显示控制和显示屏幕(CRT)等部分组成。其中,帧缓冲器图 b)是用来存储图象信息的。在屏幕上显示的图象,可按屏幕的显示精度将屏幕分割成为图 1-2 c)所示,其最小单位的小方块,称为象素(pixel 或 pixel),在水平方向上的象素的集合,称为扫描线(scan line),而扫描线的全体,构成一帧图象,称为帧(frame)。帧缓冲存储器,现在多采用集成电路的随机存取器件,每一个象素的信息,均寄存在帧缓冲存储器的对应部分中。每一象素所对应的位数(bit),对于线型图形来说,至少要有 1 位(1 bit)。当位数增加时,则可显示多种灰度等级,例如,当采用 n 位时,则可产生 2^n 种不同的灰度。一般来说,为了表示一张光滑曲面的图象,需要大约 6 位,因而可得到 $2^6 = 64$ 种灰度,理想的情况下,最好采用 8 位,也就是 $2^8 = 256$ 种灰度。以上是单色显示(黑白)的情况,图 1-3 a、b)是其示意图。

为了产生彩色图象,需要指定 R(红)、G(绿)、B(蓝)三种颜色,每种颜色分别对应的位数,如果只有 1 位,则显示器只能显示 $2^3 = 8$ 种颜色,其调色原理如图 1-3 c) 所示。如果采用 3 位,则可产生 $2^3 \times 2^3 \times 2^3 = 512$ 种颜色(如图 1-3 d)。

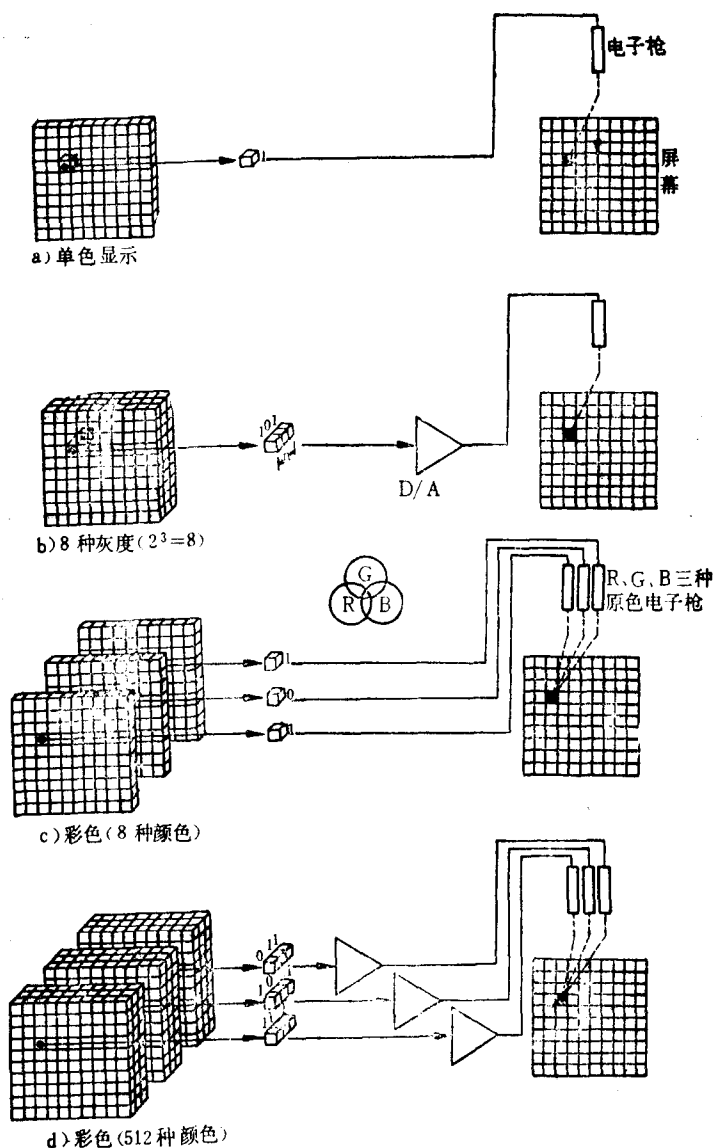


图 1-3 图形显示的几种典型设备及功能

§ 1-4 计算机图形软件标准化的基本概念与内容

一、基本设计思想

在计算机制图发展的早期,图形软件是在特定的设备上研制的,硬件也互不相同。这样研制出来的图形软件严重依赖于设备,缺乏通用性,不利于交流与推广,研制人员做着大量重复的工作,这种现象,逐渐成为图形软件研究与应用发展中的一种障碍。

随着计算机图形处理系统性能价格比的改善,以及各种图形应用软件的出现,图形软件标准化的问题被提到议事日程上来。因此,有必要明确图形软件标准化的含义。图形软件标准化意味着装置(设备)独立,即图形程序与图形输入输出设备保持相对独立性。

图 1-4 示出了非标准化与标准化图形软件系统的若干概念。图a) 表示的是两个非标准化的图形系统,设备 A 和 B 分别对应着不同的图形包,不能互换,应用程序也有不同的接口。这种程序包就是从属于设备的程序包设计应用程序的人员必须结合设备的特征进行程序设计。

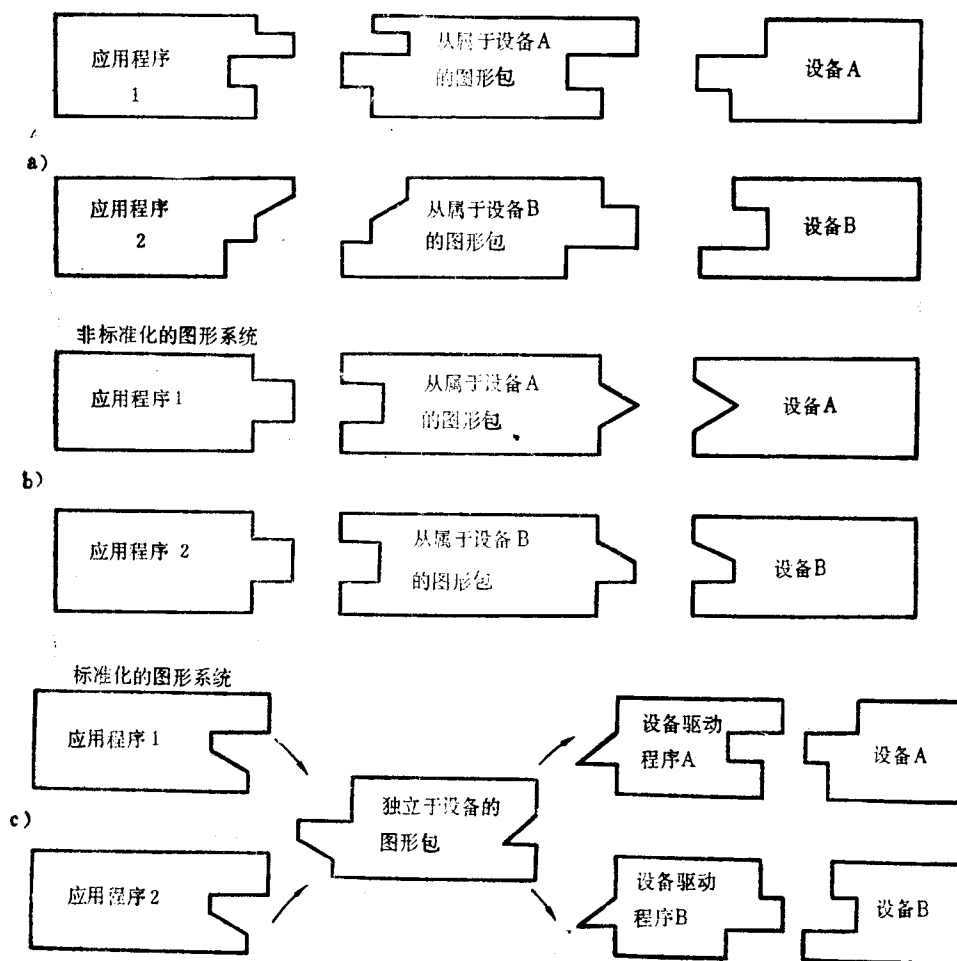


图 1-4 图形软件标准化的概念

图 b) 表示相同的程序和接口,也就是同一个应用程序,既可在设备 A 上运行,又可以在 B 上运行。在这一情况下,对于应用程序来说,虽然与设备保持相对独立,但是,这两个图形包的内容不同,不利于图形软件的迁移。

图 c) 说明在标准化的图形系统中,通过变换文件的使用,使图形包本身也与设备保持相对独立性,唯一从属于设备的部分,是在图形包和装置之间配上设备的驱动程序。因此,用户不必考虑各种设备的特性,如果要改善和加入新的设备或更换系统,只需要新的设备驱动程序

即可。

目前,发展计算机基本图形软件标准的目的是:

- (1) 实现图形应用程序的可移植性,以便在不同主机、不同图形设备之间的移植;
- (2) 建立图形软件编程方法的标准,供应用程序设计人员使用;
- (3) 实现与设备保持相对独立的标准图形包,使图形软件对设备的依赖性减至最小;
- (4) 实现图形描述的可移植性。

计算机图形软件标准的实施是相当重要而又艰巨的任务。只要实现了图形软件的标准化,便可把成熟的图形软件从一个系统方便地移植到另一系统,从而降低软件成本、避免不必要的双重应用程序的发展,使编程人员在单一的软件环境下发展计算机图形软件,同时也可图形硬件的更新指出正确的方向。

由于实际从事计算机制图的人们对标准化的迫切需要与关心,1982年国际标准化组织(ISO)正式通过将GKS(Graphical Kernel System)作为图形软件标准。

二、GKS的内容和特点

如上所述,GKS的目标是实现程序的可移植性和图形设备的独立性。为了使图形应用程序具有较高的可移植性,GKS的体系结构是以可更换性的设备驱动程序和元文件概念为特点的。因此,用户可以不必考虑各种设备的特性,若要改善或更换系统,在加入新的装置时,只需要新的设备驱动程序。GKS系统为计算机程序提供了一种图形模式和一套标准语法。

GKS的功能有10种,如表1-1所示。

表 1-1 GKS 的功能

名 称	内 容
1. 控制功能	执行打开、关闭 GKS 及工作站进入或退出活化状态,消除工作站内容等
2. 输出功能	确定输出图形的类型
3. 输出属性	设定图元的各种属性以及各图元在工作站上的表现方式
4. 变换功能	实现规范化变换和工作站变换
5. 图段功能	对图段进行创建、删除、复制以及实现图段属性(即可见性、可检测性和变换等)
6. 输入功能	对各种输入设备初始化,设定输入设备方式,确定请求、采样、事件模式的输入功能
7. 询问功能	查询 GKS 描述表、状态表以及误差表、工作站描述表、图元表等,以查找 GKS 状态值、级别和状态、工作站类型和状态以及描述表、图段状态、图元情况和出错情况
8. 元文件功能	实现对元文件的读写
9. 实用功能	用以实现坐标变换
10. 出错处理功能	用以进行出错处理

下面介绍几个基本概念,以便对 GKS 有一个初步的认识,由于 GKS 内容较多,详细内容可参看 GKS 文本。

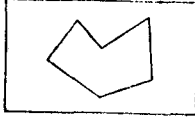
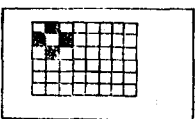
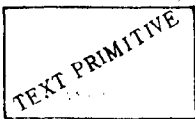
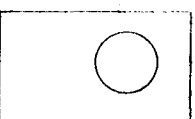
1. 图形输出

在 GKS 中有一套完整的图元(primitive),其中包括折线、多重记号、图段(segment)和字符串。有了这些图元,就可以在应用程序中用它们来构造形体,并较快地编制出交互性图形处理程序。而输出图元是 GKS 中最基本的图形信息,它使一些基本功能抽象化,这些基本功能有画线、写文字等。GKS 文本中有 6 种输出图元,即:折线、符号列、文字列、多边形区域填充、

象素组、以及广义图形元素。此外,图元的属性,由于包含了线的种类、颜色、字符的大小等等输出图元的性质,因此,它具有两种图元的属性,即几何属性与非几何属性。

表 1-2 所示为 GKS 的 6 种输出图元的名称。其中折线,标识字符(符号列与文字列)、广义图形元素为线框画图元。其余为正文图元。

表 1-2 GKS 输出图元 (6 种)

图元	名称	图元	名称
	折线		区域填充
	符号列		象素组
	文字列		广义图形元素

2. 图形输入

操作员在用设备输入图形时,GKS 是把它抽象化为逻辑输入设备进行处理。控制逻辑输入设备的属性,可在初始化时、或是在假定输入设备时加以指定。而且在设定输入设备时,可以对操作的模式进行指定。

3. 工作站

工作站的概念是 GKS 的重要概念之一。工作站是为了确保设备独立性,将多个图形设备按照一种假想的逻辑界面来进行处理,工作站是一个泛指的概念,所有具备图形输入与输出功能的设备都可以看成是工作站。以 GKS 方式把输入输出设备提供给图形应用程序,这样,就使软件人员不必顾及设备的具体硬件特性。

4. 坐标系及其变换

为了便于应用程序的使用和对输入输出设备的统一处理,在 GKS 中设置了自然坐标系(WC)、规范化设备坐标系(NDC)及设备坐标系(DC),这三种坐标系及其变换如图 1-5 所示。

5. 图段变换

在 GKS 中,图元可以图段的方式组合,并以图段为单位进行图形输出。各图段由应用程序定义,GKS 根据所指定的图段名来识别与选择。通过将各图形元素的图段化,应用程序便不再选取图段以外的图元,可将其进行坐标变换、可见与不可见、以及生成、删除等操作。

图段变换是以图段为单位的图形的平移、变比和旋转变换,是在规范设备坐标系 NDC 上进行的坐标变换。在工作站上,图段的存储空间(简称为 WDSS),即依赖于工作站的图段的存

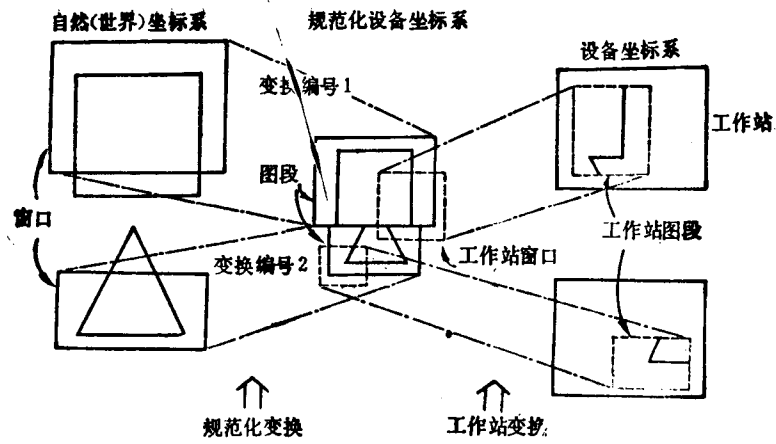


图 1-5 坐标系与坐标变换

贮空间。存放在 WDSS 中的图段是不可能转移到别的工作站中去的。

GKS 中还有一个图段的存放空间(简称为 WISS),即工作站独立的图段存放空间,这是独立于工作站的一个辅助的存贮空间,存放在 WISS 中的图段可以向其他工作站转移。这是 GKS 的又一特征。

第二章 图形处理的基本技术

§ 2-1 概 述

本章主要讨论二维图形,内容有:1) 二维图形元素(直线、圆、非圆曲线等)的生成;2) 视区、窗口与剪取的概念和方法;3) 二维图形处理中的线性、仿射、射影变换以及例图变换等的概念、性质与方法。

如前所述,本课程的研究对象是图形的生成、存储、变换和操作。作为入门工具,首先讨论二维图形的有关问题、为学习三维图形与形状处理打下基础。在许多有关计算机绘图、数控加工等参考资料和书籍中,对于构成复杂图形的基本几何元素的生成方法,都有详细的叙述,本书不拟重复,而是着重从几何算法和计算机实现的角度,选择一些有代表性的方法进行分析。例如,关于直线的生成,选择了在计算机实现方面具有特点的 Bresenham(布热森汉姆)算法,不仅方法具有一定代表性,而且也是实用算法之一。又如圆的生成也是按算法的设计思想及其发展,结合几何算法在计算机实现中遇到的问题进行分析与比较。这些方法既有区别,又有联系,结合计算机实现的可行性与难易来展开讨论。接着,在后续章节中,又从插值与逼近的角度,采用自由曲线来逼近,使读者进一步掌握分析与解决问题的方法。此外,本章还介绍了两种产生隐函数曲线的逼近方法,也都具有一定的代表性。

§ 2-2 直线的生成算法

大家知道,直线是最基本的图形元素,在计算机显示的图形中,用到了大量的直线段。除此之外,为了产生圆和任意曲线,也可以用一系列短的直线段来逼近它们。因此,如何产生直线是首先必须掌握的。

计算机产生的直线一般通过程序来实现,一个好的直线程序设计,必须达到如下几点要求:1) 直线段的始点和终点要准确;2) 所显示的线条要直,并且均匀;3) 直线的生成速度快,计算量小。这些要求也是衡量程序设计质量的一个依据。

一般来说,产生一段直线只需用解析几何中的直线方程,即可计算出直线上的一系列点,例如,当已知两个端点 $P_0(x_0, y_0)$ 和 $P_1(x_1, y_1)$ 时,可用下式计算直线上的点 $P(x, y)$:

$$y = \frac{y_1 - y_0}{x_1 - x_0} (x - x_0) + y_0$$

在具体编程时,考虑以下的算法步骤:

- 1) 输入直线的两个端点 $P_0(x_0, y_0)$ 和 $P_1(x_1, y_1)$;
- 2) 计算直线上足够多的点的 x, y 值(选适当步长);
- 3) 在屏幕上产生图形显示(或绘图仪输出图形);
- 4) 结束。

上述算法存在着这样的问题:

- 1) 当 $x_1 - x_0 = 0$ 时, $(y_1 - y_0)/(x_1 - x_0)$ 变得无限大, 因而无法实现对 y 值的计算;
- 2) 当直线与 X 轴的夹角愈大, 沿 X 方向每走一步 Δx , 则沿 y 轴方向的增量 Δy 也愈大。当夹角接近 90° 时, 则绘出的点的间隔就会很大, 打点很稀, 显示的质量降低。

为了改进上述算法, 在输入始、末端点后, 应对 Δx 和 Δy 进行比较, 找出最大的位移轴, 并以此轴为 X 轴(即满足 $0 < \Delta y < \Delta x$), 便可避免上述算法的缺点。由于 $A = \Delta y / \Delta x$ 是一个实数, 运算起来比较慢, 故不是一种实用的算法。其解决办法是采用整数处理。

下面着重介绍另外一种直线的生成方法, 即 Bresenham 算法, 是微机绘图软件中的一种实用的方法。

Bresenham 于 1965 年提出的直线生成 Bresenham 算法的基本设计思想是这样的:

首先, 找出直线的最大位移轴为 X 轴(即满足 $0 < \Delta y < \Delta x$)。令 X 轴每次只改变一步, 而 Y 轴则选择与其真值最为接近的数值(取整数值, 因为 X, Y 向的步长只能走整数步), 直线的精确路径与实际所产生的点之间的距离, 称为误差, 此项误差的量度, 是沿垂直于 X 轴的方向量度的。步长一般可取为 1, 也可取为 t 。

1. 基本算法

如图 2-1 所示, 设起点为 $P_0(x_0, y_0)$, 终点为 $P_1(x_1, y_1)$, 并令 $x(0) = x_0, y(0) = y_0$, 最大位移轴为 X 轴, $\Delta x > 0, \Delta y > 0$, 并且 $\Delta x > \Delta y$ 。

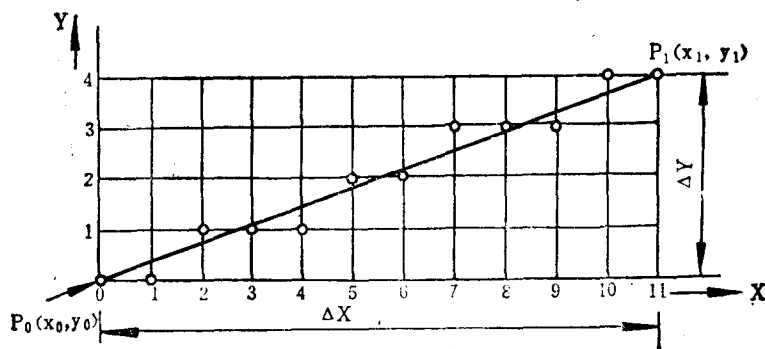


图 2-1 Bresenham 算法

用适当步长, 将 $\Delta x, \Delta y$ 等分, 如图所示的网格形状, 求出第 i 步时的坐标, 即 $x(i), y(i)$, 其中, $x(i)$ 可以用上一步的 $x(i-1)$ 来表示, 它们之间仅仅相差一个步长, 故有以下的关系:

$$x(i) = x(i-1) + 1$$

在 $x(i-1)$ 处, 直线上的点的 y 值(真实值)和 $y(i-1)$ 之差, 设为 $E'(i-1)$, 则在 $x(i)$ 处, 有