



C 和 C++

图形程序设计基础



Mark Finlay

Software by John B. Petritis



希望

科学出版社
龍門書局

73.9621
563

C 和 C++ 图形程序设计基础

Mark Finlay

著

曹 康 李增民
崔洪斌 李志诚

译

燕卫华

校



科 学 出 版 社
龍 門 書 局

1995

9610086

(京)新登字 092 号

JS113/26

内 容 简 介

本书讲述了 C 和 C++ 图形编程的基本技术。全书分为十六章,从 C 语言和几何代数基础知识入手,介绍了图形变换、裁剪、实体绘图和函数绘图,以及反混叠及动画等高级编程技术,并举例说明了编程及动画效果。本书内容由浅入深、语言流畅,是 C 和 C++ 图形编程的优秀参考书。

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

版 权 声 明

本书英文版由 M&T Publishing 出版,版权归 M&T Publishing 所有。本书中文版由 M&T Publishing 公司授权出版。未经出版者书面许可,本书的任何部分均不得以任何形式或手段复制或传播。

C 和 C++ 图形程序设计基础

Mark Finlay 著

曹 康 李增民 译

崔洪斌 李志诚

燕卫华 校

责任编辑 汪亚文

科学出版社 出版
龙 门 吉 局

北京东黄城根北街 16 号

邮政编码:100717

双青印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1995 年 5 月第一版 开本:787×1092 1/16

1995 年 5 月第一次印刷 印张: 19 $\frac{1}{8}$

印数:1—5000 字数: 437,000

ISBN 7-03-004834-2/TP·468

定价:49.00 元 (含盘)

谢 语

在很多人的帮助下,本书的出版才成为现实。首先,我要感谢编辑,感谢在整个过程中我的妻子 Nikki 对我的帮助和理解,尤其是在本书似乎出版无望的时候。除了其他许多贡献外,Nikki 还完成了大量的编辑工作,并帮助我重写了本书的主要章节。

许许多多的感谢和荣誉应归功于程序员和工程师 John Petritis。John 负责编制了本书中几乎所有的程序。没有 John 的献身精神、一丝不苟和不分昼夜的工作,书中的软件是不可能完成的,而且也不可能运行得像现在这样好。

我非常感谢 Robin Maffett 帮我绘制了许多图形。如果没有她的努力,书中的插图就会大大地减少。

我还要感谢我的技术编辑 Stephen B. Coy,他提出了许多明智的评论和忠告。同时还要感谢我的编辑 Christine 的惊人的编辑水平和耐心。

书中所有软件均用 Borland C++ 3.1 编写,本书中的文字采用 Microsoft 公司的 Microsoft Word for Windows 2.0 编写。

我还要感谢我的家人和朋友对我编著此书时的支持和帮助。

最后,我要感谢我的儿子 Sean 和 Paul 为我提出了这个项目,并始终鼓励我。

前 言

本书是为任何要用 PC 机开发计算机图形的人而写的。用户不必具有计算机图形的经验,但一定要熟悉 C 程序设计语言。

本书特别适用于初学者。对于初学者,可以从每个绘图技巧和算法后面的补充说明中受益。对于比较有经验的图形设计程序员,将发现其他图形软件包所没有的许多应用实例和工具。

本书涉及的编程技术包括:

- 以轮廓线形式和实体填充形式画圆、椭圆和多边形;
- 通过缩放、旋转和平移等变换进行图形变换;
- 用不同方式产生一个完整的绘图软件包;
- 完全交互式的图形动画制作。

读者至少需要一台带协处理器的 286 微机,以及一个 EGA 或 VGA 图形卡和显示器。源代码使用 DOS 下的 Borland C++3.1 开发环境。但是,所有软件均需符合 ANSI C 的标准,且稍加改动或根本不用改动,即可用于其他开发环境,如 Microsoft C 或 Zortech C++ 等。通过对专用显示设备例程进行少量修改,这些软件还可用于非 DOS 环境,如 MS-Windows 或 X-Windows。

目 录

绪论	(1)
0.1 基础知识	(2)
0.2 光栅图形介绍	(2)
0.3 基础知识	(2)
0.4 高级绘图方法	(4)
0.5 附录	(4)
第一章 几何学和代数学基本知识回顾	(6)
1.1 几何图形的协调	(8)
1.2 直线方程	(9)
1.3 实用函数.....	(14)
1.4 合适角度.....	(14)
1.5 关于平方根函数.....	(16)
1.6 转换成 C 程序	(16)
第二章 C 语言程序设计技术	(17)
2.1 ANSI 函数和非 ANSI 函数	(18)
2.2 头文件和模块文件.....	(19)
2.3 静态变量.....	(19)
2.4 头文件.....	(20)
2.5 宏.....	(22)
2.6 指针.....	(24)
2.7 其他参考资料.....	(27)
2.8 图形程序设计.....	(27)
第三章 图形显示设备	(28)
3.1 早期绘图发展简介.....	(29)
3.2 手工绘图.....	(30)
3.3 绘图设备.....	(30)
3.4 阴极射线管 CRT	(33)
3.5 电视屏幕.....	(33)
3.6 帧缓冲存储器.....	(35)
3.7 画直线.....	(38)
第四章 画直线和窄条	(39)
4.1 光栅坐标系.....	(40)
4.2 画点.....	(40)
4.3 绘图效率与使用方便.....	(41)
4.4 ggraph.c 模块	(41)
4.5 帧缓冲存储函数.....	(42)

4.6	调色板访问.....	(43)
4.7	画直线.....	(46)
4.8	中点选择算法.....	(51)
4.9	完整的直线画法.....	(52)
4.10	演示程序:gldemo.c	(55)
4.11	硬件帮助	(55)
4.12	更有趣的图形	(55)
4.13	本章有关函数	(56)
第五章	圆、椭圆和其他有趣的图形.....	(59)
5.1	绘矩形.....	(60)
5.2	画圆.....	(61)
5.3	快速画圆.....	(63)
5.4	用差分法画圆.....	(66)
5.5	保持适当的纵横比.....	(67)
5.6	椭圆.....	(68)
5.7	多边形.....	(72)
5.8	多边形的类型.....	(76)
5.9	绘多边形.....	(77)
5.10	曲线和折线	(77)
5.11	近似画曲线	(78)
5.12	多项式曲线	(79)
5.13	绘制更好的图形	(80)
5.14	本章有关函数	(81)
第六章	线性代数基础	(83)
6.1	仿射变换.....	(84)
6.2	线性代数.....	(86)
6.3	利用矩阵表示变换.....	(86)
6.4	多重变换.....	(88)
6.5	矩阵库与向量库.....	(89)
6.6	矩阵与对象.....	(92)
6.7	使对象绕着它的中心旋转.....	(92)
6.8	缩放问题.....	(94)
6.9	画变换后的对象.....	(95)
6.10	画其他的参数曲线.....	(102)
6.11	变换所具有的优点.....	(102)
6.12	演示程序简介.....	(107)
6.13	超出屏幕现象.....	(107)
6.14	本章有关函数.....	(107)
第七章	裁剪和开窗口.....	(110)

7.1	矩形绘图区	(113)
7.2	检查边界框	(113)
7.3	对直线作解析裁剪	(118)
7.4	多边形的裁剪	(123)
7.5	开窗口	(128)
7.6	裁剪函数	(129)
7.7	裁剪演示程序简介	(132)
7.8	实心体简介	(132)
7.9	本章有关函数	(132)
第八章	画实心图形	(134)
8.1	扫描转变	(135)
8.2	填充凸形对象	(137)
8.3	填充非凸多边形	(145)
8.4	裁剪与填充	(151)
8.5	填充图案	(152)
8.6	其他绘图技巧	(154)
8.7	本章有关函数	(154)
第九章	绘制位图	(157)
9.1	位图	(158)
9.2	建立位图	(160)
9.3	建立几何位图	(161)
9.4	根据数组建立位图	(161)
9.5	裁剪和位图	(163)
9.6	定义裁剪区域	(165)
9.7	使用位图的画笔	(167)
9.8	位图图案	(170)
9.9	位图与向量图形比较	(172)
9.10	位图演示	(174)
9.11	位图图形	(174)
9.12	本章有关函数	(174)
第十章	写文字	(177)
10.1	图形文字	(178)
10.2	带字体的文字	(179)
10.3	写字符	(181)
10.4	位图字库	(182)
10.5	矢量字库	(185)
10.6	建立矢量字库	(187)
10.7	变换位图字符	(190)
10.8	增强文字功能	(194)

10.9	演示时间	(196)
10.10	小结	(196)
10.11	在应用之前	(197)
10.12	本章有关函数	(197)
第十一章	绘制函数曲线	(200)
11.1	图形学发展史	(201)
11.2	函数绘图的常用方式	(202)
11.3	曲线的图解	(205)
11.4	画坐标轴	(206)
11.5	设置绘图比例	(212)
11.6	线性缩放	(212)
11.7	查找最小值和最大值	(215)
11.8	绘函数曲线的其他问题	(216)
11.9	对数缩放	(218)
11.10	标记曲线	(220)
11.11	绘制完整的曲线	(220)
11.12	参数函数	(223)
11.13	离散图	(224)
11.14	直方图	(225)
11.15	多条曲线	(226)
11.16	平滑曲线	(227)
11.17	本章有关函数	(227)
第十二章	计算机动画	(229)
12.1	动画的基本原理	(230)
12.2	计算机动画	(231)
12.3	动画位图	(232)
12.4	存储背景	(233)
12.5	翻动位图	(235)
12.6	动画线性画	(237)
12.7	逐位绘图	(238)
12.8	平面掩码	(240)
12.9	多图形平面	(243)
12.10	获取用户输入	(246)
12.11	动画的未来	(247)
12.12	本章有关函数	(247)
第十三章	绘制更好的图形	(249)
13.1	反混叠直线	(250)
13.2	反混叠调色板	(253)
13.3	绘制反混叠直线	(254)

13.4	反混叠的圆.....	(256)
13.5	绘制反混叠变换的椭圆.....	(259)
13.6	证实.....	(262)
13.7	反混叠技术的其他特性.....	(262)
13.8	主要程序.....	(263)
13.9	本章有关函数.....	(263)
第十四章	综合应用.....	(266)
14.1	模拟钟.....	(267)
14.2	生态系统模拟.....	(270)
14.3	更加物理化的模拟.....	(274)
14.4	小结.....	(277)
14.5	本章有关函数.....	(278)
第十五章	结束语.....	(279)
15.1	不断更新的绘图方法.....	(280)
15.2	三维计算机图形.....	(280)
15.3	标准图形.....	(281)
15.4	硬拷贝.....	(283)
15.5	小结.....	(283)
附录 A	从 C 到 C++	(285)
A.1	矩阵和向量类	(285)
A.2	图形对象	(287)
A.3	小结	(289)
附录 B	软件的安装	(290)
B.1	磁盘上的文件	(292)

绪 论

假设某公司要处理最后一个季度的财务报表,并且要求对其结果作一个总结报告,或者在刚刚收到实验室测试的粒子束实验数据,时要求在双对数坐标纸上画出各数据点,并和前一天得到的数据结果进行比较。表达这种数据最好的方法是通过图解,如圆饼图、直方图、线条图或其他图形。但是用手画这些图需要花费很多时间。

虽然我们已经认识到可以用计算机进行绘制,但以前从未有人做过。那么,怎样用图形表示这些数据呢?

图形几乎渗透到生活中的各个领域。从文件到每一本书、杂志和报纸中的广告标识符,到图或其他说明以及上述各种组合所表达的信息,图形都是人们最喜欢用的表达信息和思想的方法。

人们可形象地画图,并能快速理解复杂图中表达的含义,而用其他方法表达可能是难以理解的。今天,计算机图形被用于几乎所有需要形象化描述的操作中。读完本书后,读者将学会怎样在计算机上做二维图形。

使用计算机绘图的原理与使用其他绘图工具(如纸和笔)绘图的原理相同,都是为了用直观而有意义的方法表达和说明某种概念。

因此,到目前为止,计算机是最强有力的开发工具。使用这种工具,可以很容易地直接绘出各种图形,无论是复杂的科学数据,还是日常的商业直方图。它的灵活性和易用性使计算机成为图形世界中非常有价值的工具。除此之外,廉价的现代 PC 机使得即使是预算最紧张的商人和艺术家也能承受。

本书为需要绘制有意义图形的读者提供了绘图工具,并可避免使用相关的商品化图形软件包所产生的误解。使用本书,首先可以学到图形如何在计算机屏幕上显示和如何在计算机内存中定义图形结构。然后,可以了解作图、画点、显示正文、填充和其他相关内容。本书提出了对专业应用所必须考虑的重要的设计问题和局限性。

本书的软件都是用 C 语言开发的。本书假设读者已经具有 C 语言程序设计经验,并且可以在 PC 机上编译和执行 C 代码。本书出现的所有软件源代码均有相应软盘提供。源代码放在类属模块中,以便能在专用程序设计环境中调用。

所有源代码开发都是用 MS-DOS 下的 Borland C++ 3.1 完成。每个代码都适用于 ANSI C。设备专用程序储存在各自的源代码模块中,这些模块易适用于某些特定环境。

为了得到最好结果,推荐使用 EGA(增强型图形适配器)或者 VGA(视频图形阵列)图形卡。本书所带的软盘上有几个从各章中提取出来的演示程序,这些程序既可以用来描述特定的观点,也可以为开发自己的应用程序打下基础。请复制样板演示程序,并用来开发新的软件或增强演示功能。

正如前面所提到的,书中程序需要 80286 类型的 PC 机,至少需要 EGA 图形卡。浮点协处理器虽然不是必需的,但它可以大大提高许多演示程序的交互功能。

0.1 基础知识

在第一章中,将简要介绍数千年来人们如何用图形来表达几何关系和几何概念。但是在近代,借助于坐标系、代数和几何之间的紧密关系,人们能够开发比以前更具有实用价值的几何图形。

然后通过学习直线、圆和椭圆的方程来复习高等几何和三角学(只可惜没有例证)。这一章叙述了非常有用的计算机图形,包括表示填充面积的多边形以及用来画任意曲线和画函数的折线。最后,介绍了非常有用的数学函数(如 `sine()` 和 `cosine()`) 及其应用。

0.1.1 C 语言的简略复习

第二章复习了 C 语言及其在本书中使用的语法。因为考虑到读者很可能已经具有 C 语言的编程知识,所以这部分既可作为备忘录也可作为本书的习惯用法。对于向量、矩阵、点和其他图形的实体结构,大部分定义采用 C 语言处理对象的紧凑方式。

0.2 光栅图形介绍

虽然第二章和其他章节把精力集中在软件上,但在第三章叙述了现在已经使用将来还要使用的某些硬件知识,以及硬件对软件的影响。

通过这一章,读者将认识到光栅图形显示器和模拟显示器之间的差别,如笔式绘图仪和矢量图形显示器。在理解各种因素(如屏幕分辨率)如何影响数据的显示时,这些差别是非常重要的。

0.2.1 C 语言图形库

在复习了数学、软件和硬件基本知识后,在第四章才真正开始定义基本图形库函数。这些例程包括与显示器有关的函数(例如 EGA, VGA 或 SVGA 显示器)以及开发和显示器无关的代码所用的函数和方法。

本书介绍了基本的画点函数和在计算机屏幕上两点间画线段的画线算法。还介绍了 VGA 彩色显示器的使用,特别是如何选择及处理色彩。

第四章介绍了如何使用大多数图形卡所限定的颜色范围及如何使程序结构化,以便对不同的显示卡,无需改动程序也能产生相同的图形。

第五章采用与画线相同的算法,介绍了绘矩形、圆和椭圆的函数。然后,还介绍了如何将这些技术推广到可以绘任意曲线和形体。

0.3 基础知识

在下面各章中,将利用基本函数开发程序来处理在使用和建立计算机图形时所遇到的各种问题。

0.3.1 图形变换

我们通常需要通过某种操作在计算机屏幕上进行图形变换,如平移(translation)(将图形移到一个新位置)、缩放(scaling)(改变图形的大小)和旋转(rotation)。在第六章中,我们将开发一些必要的数学工具来实现在第四章和第五章中所定义的各种几何实体的绘图操作。

在探索了某些线性代数方法后,读者就会明白每个变换是怎样使用线性代数 3×3 矩阵合并成单一操作的。然后,矩阵变换被合并到文件模块中,用来提供简单高效的图形变换方法。第六章介绍了各种图形是如何变换成另一个相关图形的,如圆在一个方向缩放就形成椭圆。

0.3.2 裁剪

当需要绘制多个图形及多条函数曲线时,无疑会遇到一些特殊图形。这些图形线需要局部裁剪屏幕部分图形,至少需要裁剪图形的坐标轴标尺。第七章介绍了裁剪(Clipping)的概念,这种方法可以保证只绘出在屏幕上您想让其出现的那部分图形。

裁剪尤其能使图形避免重写坐标轴标尺和图形不可读现象。本书提出了两种基本方法,第一种方法是沿一个裁剪区选取一个图形对象(如多边形),并产生一个保证完全位于区域内的多边形。然后,将这种方法推广到相对于一个多边形来裁剪另一个多边形。第二种方法是在绘图函数内部使用显式检查以保证不绘在屏幕所选区域之外。两种方法各有优缺点,这部分内容和何时应该使用哪种方法以及何时使用比较简单的方法将一块讨论。

裁剪的主要用途之一是支持选取窗口(windowing),选取窗口就是将屏幕的一个矩形区域定义为要操作的图形区域。基于窗口系统的所有软件,包括 Microsoft Windows,进行绘图时都必须使用变换操作(transformation operation)(在屏幕上移动所有对象)和裁剪操作(避免将数据写入其他窗口)。第七章将说明这种操作如何实现,尤其是这种需要为什么滞后于基于窗口的环境。

0.3.3 实体对象

前面几章详细论述绘制几何图形边界线或轮廓线的方法。第八章介绍如何将绘图方法扩展到用实体颜色和图案来填充对象。第九章论述了如何生成用来表示图案或图像的位图,以及如何使用位图来表示图符,图符可提供快速绘制固定图形实体,如光标和简单的文本。位图也可以用来储存线图图案,包括虚线和点划线或者更复杂的在大多数计算机着色程序中看到的刷涂图案。

第十章介绍了绘制更复杂文本的各种方法,尤其是怎样绘制基于位图的文本和基于向量的文本。位图文本一般可以很快绘出,但是文本转换比较困难,尤其是任意缩放和旋转变换。

同时,还介绍了某些与基于位图和基于向量的文本相关的显示图形(这里的意思是图形具有奇异的或错误的形貌),并且介绍了如何用更高级的图形软件包处理这种现象。

0.3.4 主图绘制

第十一章讨论了与绘制数学函数图形有关的问题。函数可用数学方法表达,如多项式或

给定用自变量(通常与 x 坐标相关)和因变量(通常与 y 坐标相关)表示的数据文件来表达。

函数曲线的变化类型实际上是无限的,决不只限于本章提出的类型(事实上,有许多书籍专门研究这个课题)。但是,第十一章提出了许多涉及该课题的问题以及如何按特殊需要访问它们。本书提出了几种方法可用来创造性地利用颜色以及采用清晰而有意义的方法表示多次函数。本书还介绍了怎样制造“伪”数据以及如何产生令人费解的或不完整的曲线。虽然从来没有人故意这样做,但有人却常常有意(为了过分强调数据的特殊方面)无意地(因盲目地选择了专用绘图软件包的缺省值)这样做了。只有注意这样的潜在失误,才能保证表达总是高质量的。

0.4 高级绘图方法

第十二章介绍了最常用的动画制作方法。前几章曾介绍过而且后面还要介绍前几章演示程序中使用的一些动画技术。本章总结了用光栅图像完成动画的各种操作以及逻辑函数,尤其是 XOR,如何采用近乎魔幻的技术在动画中发挥作用。通过使用矩阵变换,可以掌握如何借助于简单变换进行对象平移,使对象能在屏幕上走动,来完成复杂的缩放变换以及看上去比真实对象更复杂的旋转操作。

0.4.1 反混叠技术

作为最后一个主题,第十三章介绍了清除光栅图像中常有的人为痕迹问题。术语“反混叠”(anti-aliasing)涉及到消除不理想特征的整个技术领域,最引人注意的是与坐标轴成一定角度的直线图上出现的锯齿状。

反混叠技术是过去 20 年来有大量开发记载的一个复杂问题。第十三章重点讨论了有效消除画直线和圆时出现的锯齿边的最新先进技术,并论述了如何有效地将这些技术和其他图形模块合并在一起。

0.4.2 几个有趣的演示程序

有了前几章工具程序的基础知识,通过研究第十四章的样板程序,就能掌握如何联合利用这些工具编制有趣的交互程序。第一个例子是一个时针、分针和秒针可转动的表盘,第二个例子是使用牛顿运动定律模拟在封闭容器内螺旋运动的粒子。第三个例子是一个在计算机屏幕高级动画程序中,鱼和鲨鱼为了生存而互相攻击的演示程序。

0.5 附录

本书附录包括计算机图形学的一些其他论题,并提供了一些很有用的参考资料。附录 A 讨论了一些 C++ 提供给图形程序的论题和特性。附录 B 包括封口软盘上的软件名单和使用 make 或 Borland project 文件在机器上建立软件的指导说明。

0.5.1 总结

本书对计算机图形程序的开发只是初步的,还有许多不同的软件 and 标准不在本书讨论

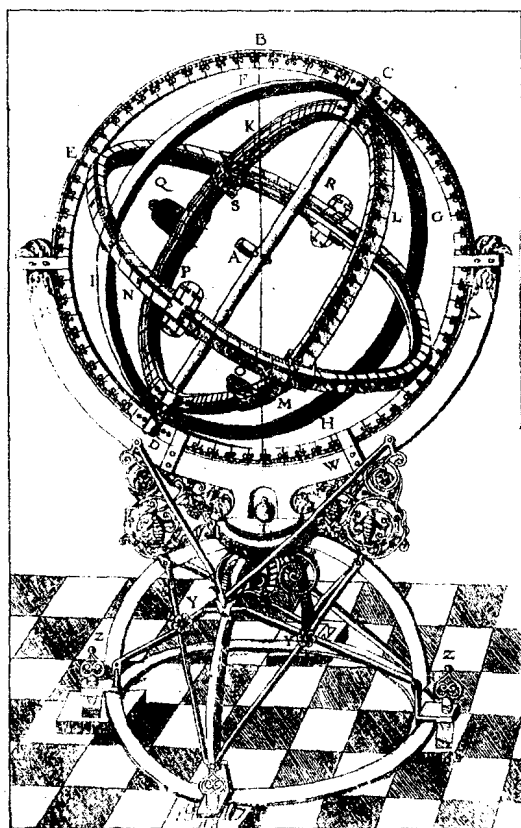
范围之内。从 Microsoft Windows 到工作站 X-Windows 系统再到 Macintosh 和许多商业软件包内部的绘图软件包,通常都难以知道从何处着手。但是,所有这些软件包都具有类似于本书中出现的软件包的普遍特性。

图形软件包不仅提供在屏幕上选择颜色和画线调用,而且还提供许多更复杂的绘制其他实体图形的调用,如画圆和多边形。第十五章将进一步研究各种图形标准和计算机图形学的未来发展动向。

书中所有程序均设计成交互式和带理解提示的,这种方式易于理解如何产生图形描述以及如何产生日常所见到的各种图形显示。使用本书所提供的源代码,可以开发出有趣的图形序列和操作者自己的图像。随着进一步深入研究,可以将这些模块扩展以满足特殊需要。

用计算机绘图的主要优点之一是可以获得既可交互式又看起来十分有趣的显示效果,只要考虑到非常丰富的大量的计算机游戏,即可体会到这一点。最佳游戏软件的主要特征是能够提供高质量的图形画面。本书的目的就是给读者一个初步知识,以便理解这类图像是如何产生的,读者还可自己实际动手绘制这些图形。这些软件是非常有趣的,继续研究计算机图形介质肯定会变得更加有趣。

第一章 几何学和代数学基本知识回顾



观像仪是一种非常美观非常精密的古代天体观测仪器,它利用十分简单的几何原理,利用对星体在空中的运动规律进行的天文观测所获得的资料来确定某天体的方位,是一种十分有效的工具。利用该仪器所进行的测量其结果是直线、圆以及球体之间几何关系的简单函数。

建立像观像仪这样的仪器需要具有坚实的几何和代数基础。一旦对这些基本学科有了深入理解,就可着手对所研究的各种现象建立自己的模型。这些模型可以是机械模型,如观像仪,也可以是纯概念模型。无论是哪种情况,绘图是表达观点的主要方法,而计算机图形学则是建立图形的一种强有力的工具。

在本书开始首先回顾一下几何学及代数学知识,一旦掌握了几个主要概念,就可以绘制对象的二维图形,如绘制直线、圆以及椭圆等。

几何学的研究引出了图形学。人们借助于绘图来观察和解决问题是客观的需要。虽然绘图方法在时代发展过程中发生了巨大变化,但绘图贯穿整个历史发展过程。

最早人们绘图所使用的最简单的工具是纸和笔。三角尺用来画直线,简易圆规用来画圆。许多古典的几何定理都是用这两种工具推导出来的。

从技术上讲,许多几何学与代数学的理论研究并不需要绘图,但是因为人脑具有精细的形象化处理功能,所以绘图比用符号能更多地表达与一个概念有关的信息。换句话说,一幅图形胜过一千句话。人是形象化描述事物与表达事物的主体,而图形学则是提供形象地传播复杂思想和概念的手段。

推动对精确绘图方法需求的三大主要学科是天文学、建筑学和地图制作。因为精确的天文测量需要深刻理解几何学定理,所以借助详细设计的图来构造精密测量仪器的能力以及观察某些天体相对于其他天体位置的能力就变得更为重要。

在整个历史过程中,为帮助绘图以及测量行星和恒星的复杂运动情况,人们研制了许多精巧灵敏的机械装置。例如,望远镜需要有一个复杂的钟表和齿轮装置,它要随着地球的旋转日夜跟踪目标。虽然测量和观察仔细,但这些装置仍需改进,以便使装置能在固定地点用望远镜跟踪所观察的目标,获得更高倍数的、更详细的视图。这仅仅是一个使用数学描述过程的机械模拟的实例,在这种情况下,这个过程就是行星的运动。

现在再看一下人工呼吸器,这是一个令人惊奇的玩具。当一个圆在另一个圆内旋转时,它跟踪圆周上固定点的运动。用这种简单装置所画图形的变化是非常惊人的。如果给这种装置添加一些附加的特征,如用椭圆代替圆,那么图形的变化可能是无穷无尽的。

这种人工呼吸器与古代天文学家用来模拟地球周围大气运动的装置很相似。在这种装置的模型中,任何物体均在中心球形表面的圆周上运动。还有些观察到的运动格外“独特”,如逆向运动(行星穿过天空轨迹的明显的逆向运动),天文学家建立了行星在一个较小圆周上围绕无形圆心转动的非常精密的模型,无形圆心在圆周上围绕地球转动。这实际上是月球绕行星运动的合理近似。古代天文学家观测这样的运动需要很高的洞察力,极好的数学知识以及良好的绘图和机械设计技巧。

正像大多数建筑师的忠告那样,精确绘图是任何建筑或结构设计的关键。图形早就被公认为是设计必不可少的手段。精确的机械制图可以帮助设计者观察最终结构怎样看起来美观,还可以提供要制造这样的结构将需要多少材料的衡量方法。这些图纸还可以用来进行辅助结构分析,以便了解这种结构的使用稳定性。然而,从早期机械制图的研究到现代材料科学和机械工程,我们已经走过了很长的路程。

为了提高绘图效率,建筑师采用了许多美术和几何学的技术,提出了现在被称为绘草图的方法。在较早年代,这样的图实际上是艺术家的再现。当这些图用于观察最终结果时,通常因为太不精确而不能用于实际测量。用于测量的地方在图上做上标记,这就需要所有需要严格测量的地方均要有各自的标记,因为不能保证所有元件绘图比例均合理。由此可以想像获得一张“新”的图纸是多么困难。

今天,大多数人都认为具有按适当比例绘制草图的能力是理所当然的,也就是在绘图时,图中所有尺寸均要按与建筑物或结构的最终尺寸相关的比例正确绘制。然而最基本的是目前快速而简易地复制图的能力。

使用计算机,改变并获得一张最终的新图纸是非常容易的。正如用廉价实用的文字处理