

如何用 C 语言创建高级 可视化图形程序

张 昭 丁 东
李 伟 何 林
刘 畅 江 波 编
万 军 刘 强



北京希望电脑公司

如何用C语言 创建高级可视化图形程序

张	昭	丁	东	
李	伟	何	林	
刘	畅	江	波	编
万	军	刘	强	

北京希望电脑公司

前 言

本书是向你提供在个人计算机（即PC机）上如何用C语言写有关可视化图形（visualization graphics）的编程技巧。

如果你确实对C图形程序设计感兴趣，那你正好赶上好时候。产业界权威人士声称，九十年代将是个人计算机图形学的十年。

计算机图形学的编程包括了各种训练，如三维实体造型和真实感图形生成、模拟和动画、图象分析和润色、计算机视觉和图象处理、图象形变与中间帧生成（即morphing和weening）等等。显然这些工作有助于我们感觉和理解，其结果便是计算机可视化（Computer visualization，又称计算机直观法）——用计算机进行直观地思考。

计算机可视化增强了我们解决复杂问题的能力，从而实现创造性思维。可以说可视化改革并翻新了工程和建筑、工业设计和制造、视觉艺术及设计、医学、物理学和生物学、娱乐和广告业。计算机可视化已经唤起了第二次计算机革命。九十年代热门的多媒体（multi-media）和虚拟现实（virtual reality）莫不包含了一种极具生命力的技术。

具体说，本书的目就是帮助读者用C语言最大程度地在个人计算机上开发商业、科学、医学、工程设计等方面大的图形应用程序，编写娱乐的图形软件。

书中附有大约1MB的源代码（超过1万行的程序清单），同时还有约2百多幅画面、照片，非常适合在个人计算机上进行图形编程的各种层次的人员参考。

如果你刚刚涉及C语言编程，你可以使用书中的程序例子进行训练；如果你是中级或高级程序员，本书可以帮助你开发有效更具竞争力的软件，同时还介绍了一些先进的高性能技术。当然本书也对正从事或打算从事图形软件开发的管理人员、程序人员、系统分析员极为有用，对工程师和科学家也有重要参考价值。

本书的宗旨是实用，为此里面附有大量的实例，而且没有任何玩具程序。每个例子，每个演示程序，每个代码段，每个算法都是可用的。从这个意义上说，本书又是一个强有力的工程工具箱。

本书的一个重要特点，就是要运行的演示程序的全部代码和全部有关原理都在书内可以找到。为此书后附有一个代码工具箱，这种多模块的C语言编程方法使建立一个易于理解的高级复杂程序成为可能。

本书的所有程序代码都用个人计算机上极为流行的Turbo C和Quick C完成。

我们希望本书能使各种读者各取所需，从中了解到计算机可视化图形学的重要应用，体会到使用C语言进行高级图形程序设计的奥妙。

目 录

前言

概述

第一部分 可视化的编程技巧

第一章 编程基础	(13)
第二章 色彩编程	(34)
第三章 位图编程	(66)
第四章 动画编程	(87)
第五章 鼠标器和数字化板编程	(98)
第六章 语音和声音编程	(109)
第七章 前端界面程序设计	(115)

第二部分 物体的可视化方法

第八章 三维空间编程	(150)
第九章 LAMBERT 明暗处理	(162)
第十章 Couraud明暗处理	(197)
第十一章 消除隐藏面	(203)
第十二章 程式动画	(210)
第十三章 运动动画	(241)

第三部分 基于图象的可视化

第十四章 图象处理	(259)
第十五章 计算机视觉	(270)
第十六章 Cel动画	(274)

第四部分 附录

附录一、如何运行例子程序	(291)
附录二、如何选择和使用其它图形库	(299)
附录三、如何捕获程序中运行时错误	(313)
附录四、用扩展内存和扩充内存写图形程序	(319)
附录五、用C++增强图形系统	(320)
附录六、高级应用	(322)
附录七、工具包源代码一览表	(325)
附录八、本书涉及的C选摘	(395)

附录九、图形适配器与色彩调色板	(405)
附录十、演示程序(代码段)、算法(功能)对照表	(409)
附录十一、键的ASCII码和扫描码(十六进制)	(413)

第五部分 其它

术语汇编	(414)
源程序代码一览表	(428)
几种3D商品	(430)
商标注释	(431)

概 述

G.Domik在New Results and New Trends in Computer Science一书中谈到了可视化(Visualization, 也译直观法)在数据理解中的作用^①, 这是从计算机科学的新成果及新趋势角度说的。随后K.W.Brodli等又针对科学可视化的技术与应用写了一本专集^②, 系统地描述了可视化的出现、技术原理及产品应用(特别提及了九十年代热门的高技术Virtual Reality)。

那么, 什么是可视化, 它又有什么作用呢? 简言之, 可视化就是把各种繁杂的数据转换成直观的图形和图象, 从而有利于人的理解和接受。可以说可视化图形学(Visualization graphics)的出现是第二次计算机革命, 这种新涌现的软件技术代表了使用计算机图形学(computer graphics)的一种新方法。

利用图象进行思考, 使我们更易模拟现实世界或探索抽象的概念, 甚至去遨游幻想的王国。使用可视化做为思考的工具, 有助于我们去解决问题, 实现新的设计和处理, 去推测不同的选择、不同的结果及不同的条件。我们的心是面向图形的(graphics-orient-ed), 这一点谁也不能否认。

计算机可视化(或称直观法)就是用计算机进行直观的思考。更重要的是它极大地增加了人的能力, 扩大了人的思考范围, 计算机本身比我们更具有可视化的原始机制。首先计算

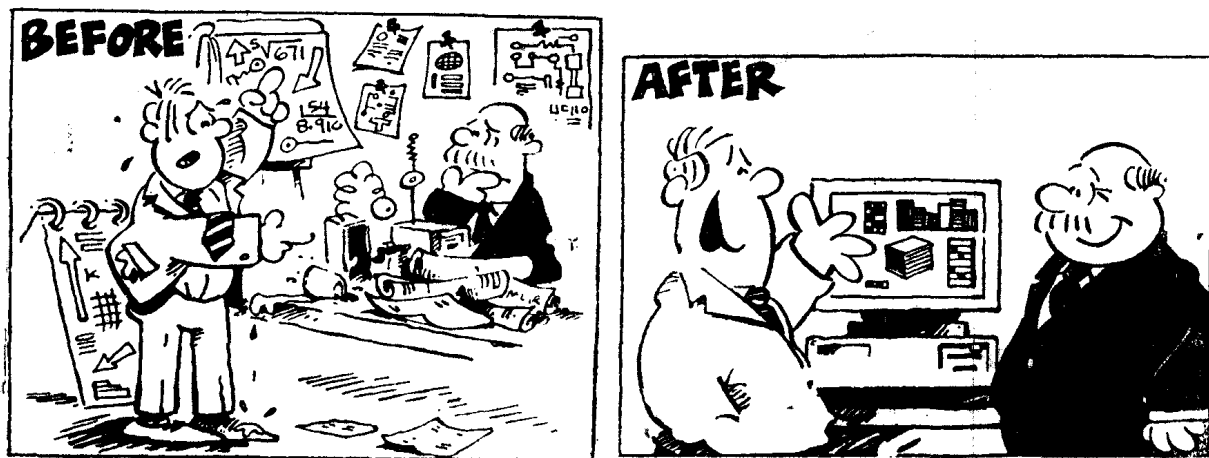


图 1 可视化的效果(漫画)

机不会感到疲乏, 可以生成一百次图象, 而每次的结果都一样地正确; 同时计算机也是一位细致的大师, 无论生成一幅图象的算法多么长, 数据多么复杂, 它都一直兢兢业业地去做; 计算机又是一位客观的合作者, 人进行可视化工作往往具有更多的主观处理, 而计算机生成的可视化图形可以消除人在计算上的局限, 让我们自由地进行主观的视觉比较。正是计算机处理了所有的客观的成分, 我们才能集中精力去做我们更想去做的事情。另外, 计算机还是一个有效的和极具说服力的通讯媒介, 无论可视化成一幅静止的图象, 还是一套动画序列, 图片的含义往往更深。

一幅图象可以克服语言障碍和专门的术语。技术人员、设计人员和工程人员可以利用可视化图形来解释复杂的信息，从而使管理人员、执行人员、客户、消费者及其它决策人员更容易理解。

人类通过观察去了解世界占总摄入量的百分之九十。可视化将是今后若干年内一种强有力的工具。

个人计算机（即PC机）已经是文本材料的强有力的处理器，如果我们再加入图形，那么就会戏剧性地至少提高百分之二十五的有效性。另外一种交互的可视化技术也使用户成为积极的参与者。如漫画中的两幅图形象地表达了计算机可视化的作用和效果。

这里还有五幅图说明了本书的内容，它们分别选自第九、十二章、十三、十四和十六章。

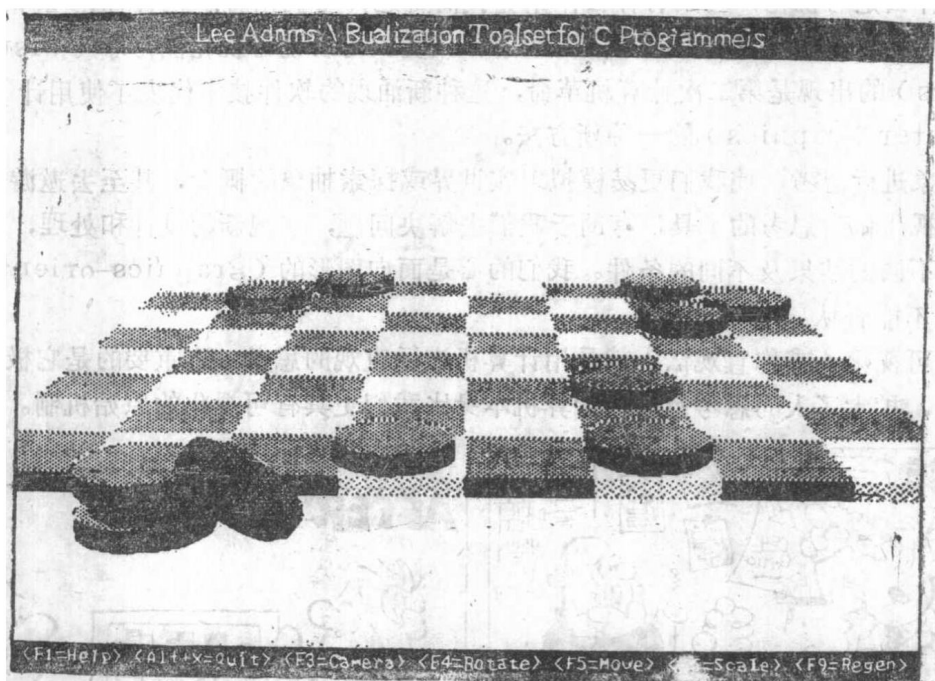


图 2 (a)

(a)图是存在于计算机内存的三维环境物体的可视化形象。

(b)图是一种计算机算法的直观化，图象表明了正在运行中的排序过程。

(c)图表示了运动动画（或称全动作动画）。物理定律，包括碰撞可以通过数学公式在三维环境中进行模拟。

(d)图演示了中间图形生成。计算机可以利用最左最右面的关键帧来产生中间的图象。

(e)图显示了程式动画、它可以结合计算机和适配器的能力来体现细微的形状、颜色、纹理位置等空间关系。

可视化图形有极为广泛的应用领域，可以说计算机可视化正在工程建筑、视觉艺术和设计，医药和生物科学、娱乐和广告、工业设计和制造、机器人和运动控制、数学和天文学等诸多领域产生革命性影响。

在医学领域内，计算机可视化利用X光层面扫描（tomography）进行内部疾病的诊断。

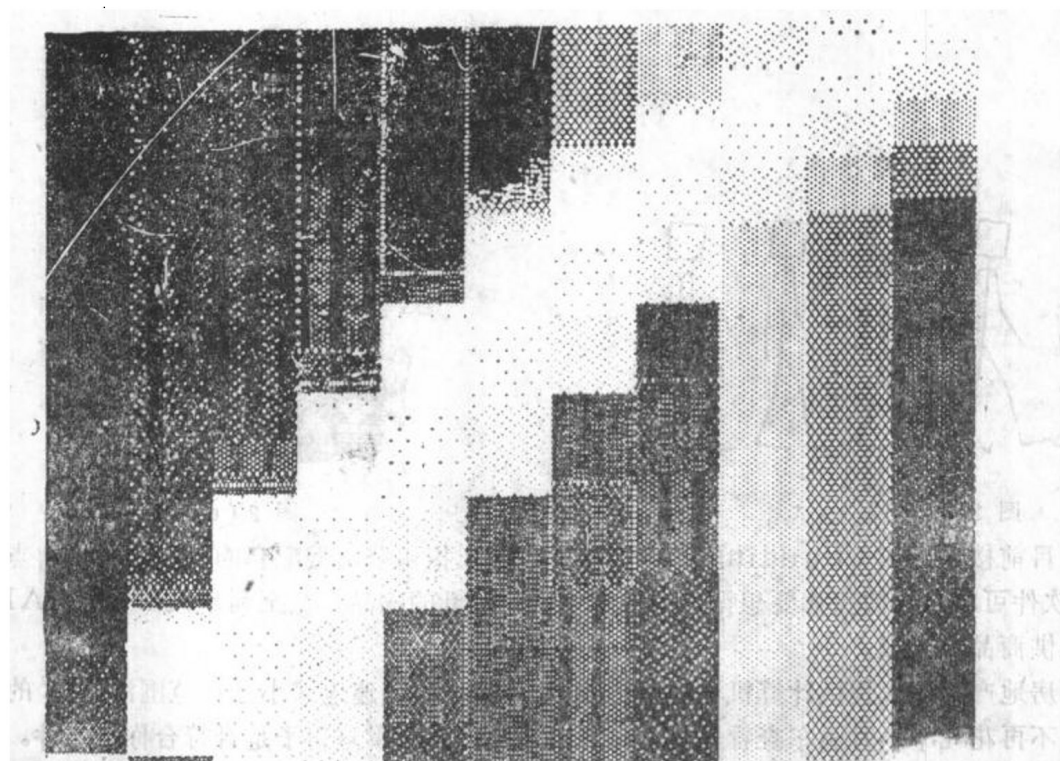


图 2 (b)

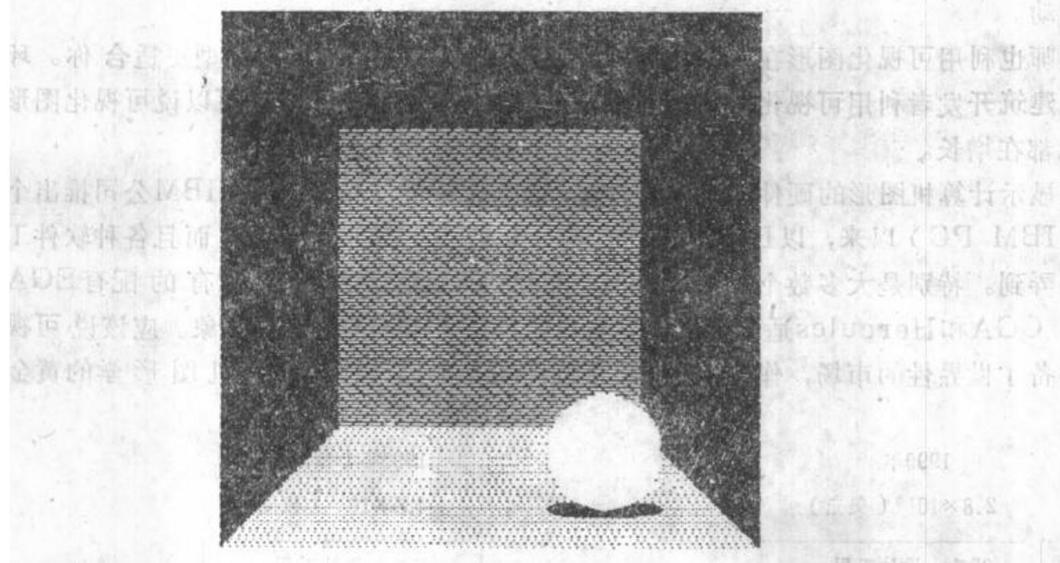


图 2 (c)

在物理和化学领域，科学可视化利用计算机生成能量、电子、场和数学公式的三维表示。

工业设计师、建筑师及工程师可利用计算机可视化准备二维布局、三维原型，进而帮助他们进行最佳的设计选择。

在广告和娱乐领域，计算机可视化能生成以假乱真的演员、产品、动画序列。

当然，计算机可视化并不局限在这些高层职业上，事实上它已渗透到我们每个人的生

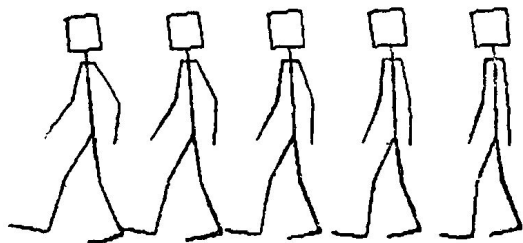


图 2 (d)

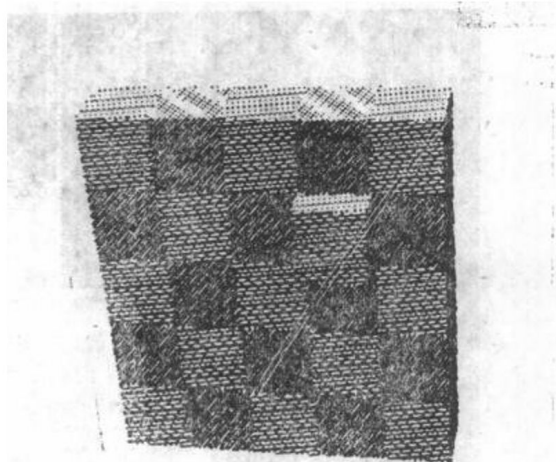


图 2 (e)

活当中。目前楼层商店 (Building supply store, 指和公寓楼配套的商店) 提供售点显示服务, 软件可以帮你购买你要想得到的各种形状或尺寸的物品, 甚至利用人工智能 (AI) 来为你提供商品系列报。

许多房地产经纪人提供计算机化的视觉数据库, 使你能快速地缩小选择范围满足你的购房要求, 不再花几个小时驱车查看房间, 你可以通过画面看看这房子是否符合你的条件。

屋内设计师——特别厨房设计人员——正开始利用个人计算机 (即PC机) 让你看看厨房是否真正适合于你。它可以很容易地把壁厨从一面墙移到另一面, 而无需在设计完或施工过程中进行改动。

发型师也利用可视化图形在个人计算机 (即PC机) 上来显示哪种发型更适合你。环境规划师及建筑开发者利用可视化图形进行各种图示来计划他们的工作。可以说可视化图形的应用每天都在增长。

现在显示计算机图形的硬件已有很大发展, 微型计算机自八十年代初IBM公司推出个人计算机 (IBM PC) 以来, 以Intel芯片为主的PC机已发展到80486机, 而且各种软件工具也很容易弄到。特别是大多数个人计算机都装配有VGA图形卡, 即使现有的配有EGA、MCGA、CGA和Hercules适配器的计算机也能提供可以接受的屏幕图象。应该说可视化图形已具备了世界性的市场, 作为一个C图形程序员, 你已处在了计算机图形学的黄金时代。

1990年	1995年 (预测)
2.8×10^{10} (美元)	4.5×10^{10} (美元)
30% 设计工程	25% 设计工程
25% 科学和医学	25% 科学和医学
20% 图形艺术	30% 图形艺术
10% 可视化通信	9% 可视化通信
10% 建筑设计	9% 建筑设计
5% 制图	2% 制图

图 3 市场分析

图3 给出了有关可视化图形的市场分析。

前面,我们谈到了计算机可视化及其应用。此外本书还有以下若干特点。

- 硬件和软件的兼容性。

书中的程序全都用Quick C和Turbo完成。强有力的预处理伪指令#if、#elif、#endif在源码中的使用确保了百分之百的语言兼容。我们可以参见第一章的有关讨论。

每一个演示程序中都有一个特殊的自动检测模块,这使得程序在启动时能支持你的系统上可能获得的最好图形。大多数演示程序支持所有流行的图形适配器:VGA、EGA、MCGA、CGA和Hercules。

和具体图形库(通常是Quick C的内部图形例程或Turbo C的BGI接口)相关的源代码主要限于附录七中的工具包模块。这同时意味着演示程序转到其它的C编程器,如Lattice C、WATCOM C、TopSpeed C、High C等相对容易。更重要的是多模块编程方法很容易使代码转向第三家图形库(如Essential Graphics、Meta WINDOW、Meta WINDOW/plus、HALO、GFX、TopSpeed's内嵌图形、WATCOM的内嵌图形、Flash Graphics等)。

- 程序清单和代码段采用一致的格式。

按下图的格式安排的代码可以使源码更易于理解。注意,每个程序清单和代码段都标上了行号,这也是从便于阅读角度考虑的。如果你想录入这些程序,切记不要敲入这些编号。

```
1201
1202  LOAD ONE FRAME OF ANIMATION SEQUENCE FROM DISK
1203
1204  */
1205  static void zloadFrame(char * Name, char far * Blk,
1206                        int FrameNum, long Bytes) {
1207      /* On entry, Name must be an 8 character filename. FrameNum
1208       must be an integer in the range 1 to 999 not exceeding the
1209       number of frame files actually present on disk. Bytes must
1210       be the size of an individual frame file. Blk is the name
1211       of the target graphic array in the far heap. */
1212      FILE * frame_file; /* data stream for disk I/O */
1213      char filename[15]; /* full filename to load */
1214      int Radix=10; /* base 10 numbering system */
1215      char frameCode[4]; /* coded portion of the filename */
1216      char fileExt[5]=".V01"; /* denotes a VIBRANTvision display file */
1217      char * CodePtr; /* pointer to coded portion of filename */
1218      if (FrameNum<1) return;
1219      if (FrameNum>999) return;
1220      CodePtr=itoa(FrameNum,frameCode,Radix); /* convert to string */
1221      strcpy(filename,Name); /* grab first 5 characters of name */
1222      if (FrameNum<10) { /* if need to pad with two zeros... */
1223          strcat(filename,"00"); goto PAD_DONE;
1224      } if (FrameNum<100) strcat(filename,"0"); /* or pad with one zero */
1225      PAD_DONE:
1226      strcat(filename,frameCode); /* append frame number */
1227      strcat(filename,fileExt); /* append filename extension */
1228      frame_file=fopen(filename,"rb"); /* open file for binary read */
1229      fread(Blk,1,Bytes,frame_file); /* read it */
1230      fclose(frame_file); /* close it */
1231      return;
1232  }

0518  outtext(Text00);
0519  setcolor(0);
0520  setusercharsize(1,3,1,3);
0521  switch (mode){
0522      case 2VGA_12n: moveto(210,y_res-23);break;
0523      case 2EGA_10H: moveto(210,y_res-29);break;
0524      case 2EGA_EH: moveto(210,y_res-23);break;
0525      case 2MCGA_11K: moveto(210,y_res-23);break;
0526      case 2CGA_6H: moveto(210,y_res-23);break;
0527      case 2HERC: moveto(210,y_res-41);break;
0528  outtext(Text01);
```

行号

函数的图形标题

注释

局部变量(缩进)

缩进的续排内容

目标行标号

缩进的switch()块

图 4 源代码格式

清单中的代码块是缩进的，从前面的代码行续下来的内容也是缩进的。Quick C和Turbo C编辑器被设置成用二个空格分隔代码的缺省方式。

完整长度的演示程序都以规范的标题块开始。如下图所示，这种方框中包含了关于源文件名字、代码的目的、源文件列表的重要信息。另外，它还包括使用完成的程序的信息。如果支持命令行变元，它们也在这里解释。更详细的信息可参阅第一章和附录一。

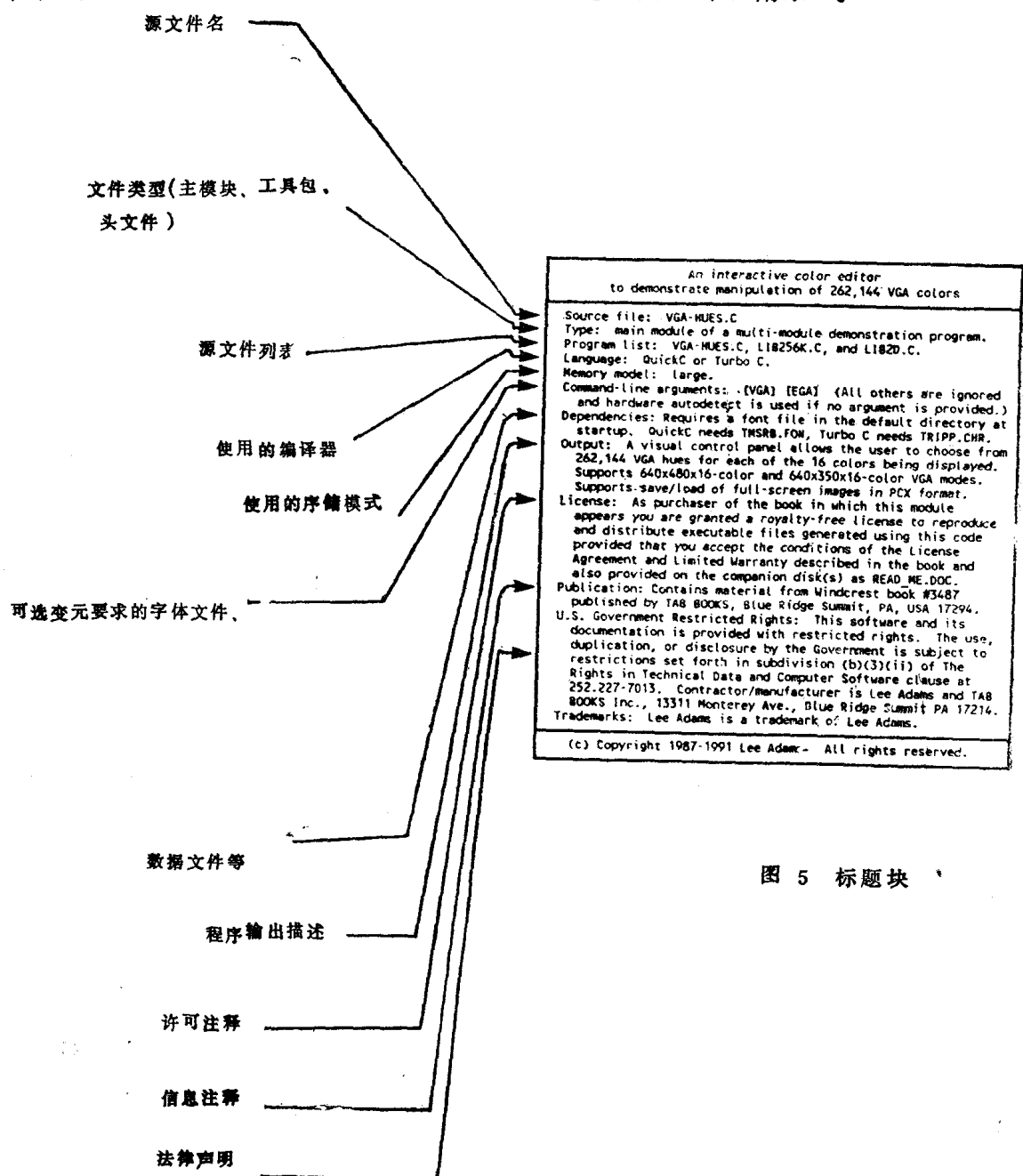


图 5 标题块

本书的代码段和程序清单都保持统一的编程风格，如下图所示。

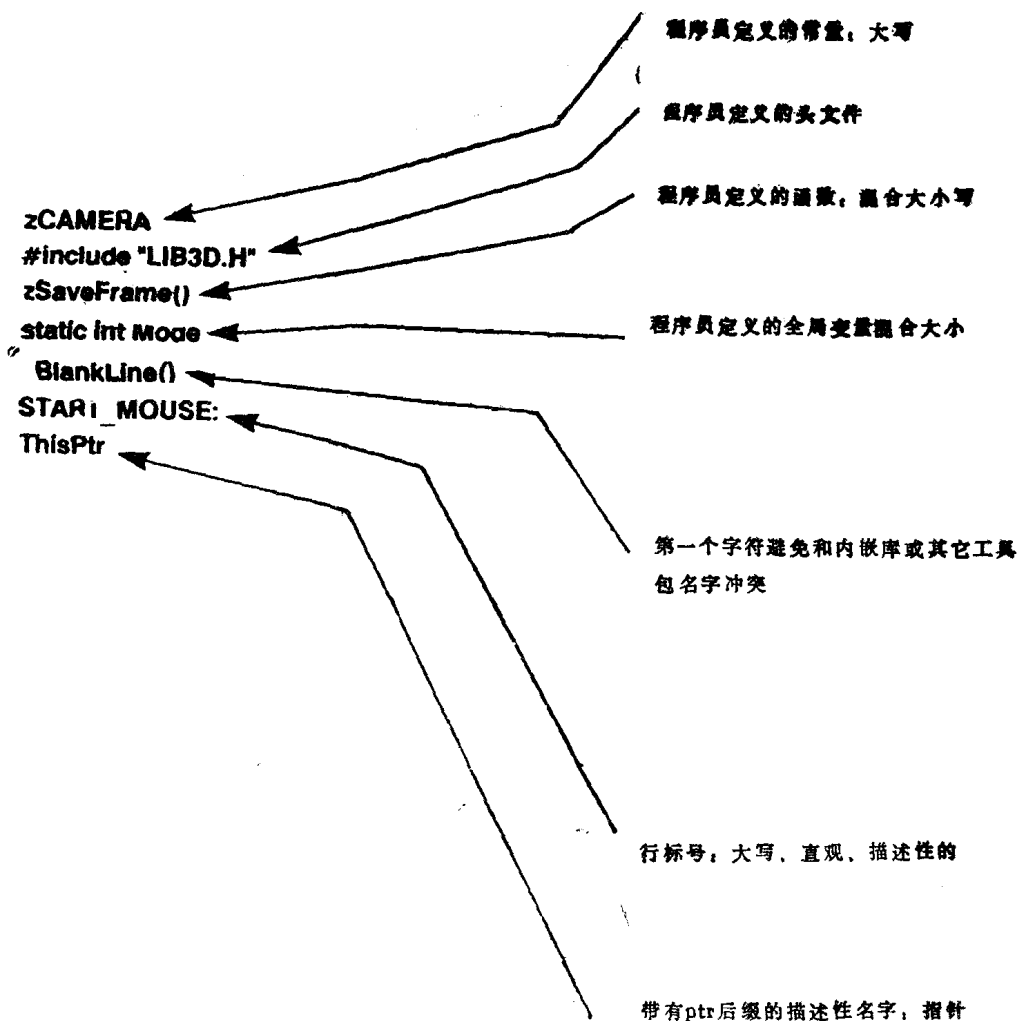


图 6 编程风格 I

每个常量的名字全都大写，第一个字母用Z来帮助避免同已有的编译器库或第三家库中的常量名字冲突。其它如变量名、程序函数名和指针变量名也都本着类似的原则。

编程风格还体现在C语言流程的控制处理上面。尽管C语言赋予了强有力的语句，但它也很容易使人滥用它的流程控制能力。

下图中while()语句产生难于调试的循环，特别是在控制条件块执行的测试表达式在运行时不完全可预测的情况下更是如此。这种类型的循环最好使用简单的yes-no条件。中间的for()语句由于使用不必要的复杂的初始化条件，测试表达式和修改表达式，也使它难于理解。这种类型的循环最好使用简单的计数条件。下面的硬代码循环(hardcoded loops)则容易同汇编语言相接，从而有利于速度和大小的优化。

C语言的Goto指令常常用于构造简单的循环，特别是在while()和for()生成的代码难于理解和维护的复杂情况下。可以说循环的goto风格经常用在减少代码复杂性和增加可读性的场合。甚至CASE工具也开始承认goto语句了。goto语句在生成结构化、有效的可维护的C代码的场合能扮演重要角色(参见第一章有关内容)。

本书中的每个程序清单都伴有一幅屏幕打印。这显然有助于解释和图解正在讨论的内容，

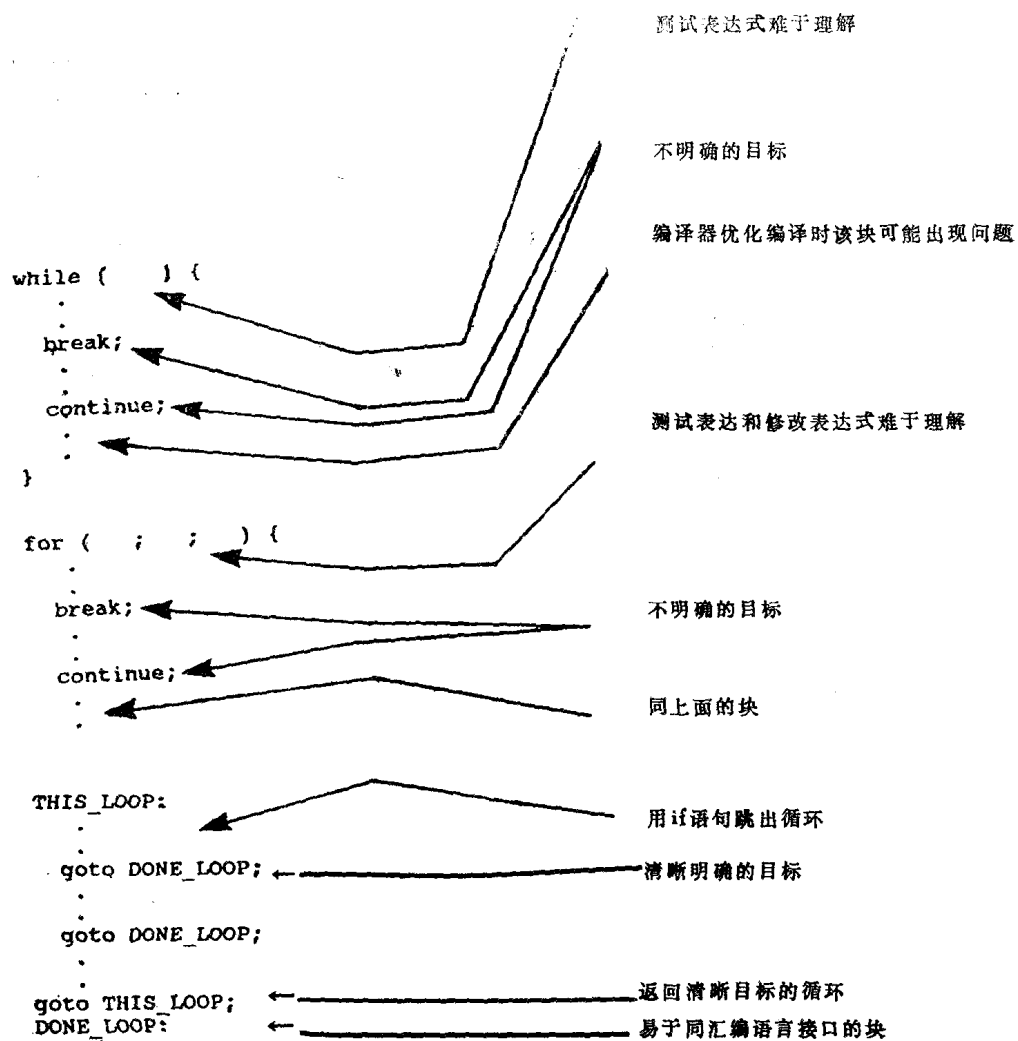


图 7 编程风格 I

同时也利于读者在自己的机器上进行比较。图8和图9分别表示了这种思想。

除了演示程序、代码段、伪代码、文字讨论外，本书还有二百多幅插图、卡片、图解和照片。许多声称是教如何进行图形编程的书，本身却缺少必要的图形，很难想象他们是怎样教的。

我们所了解的内容百分之九十多是通过观察、看、视觉经验获得的。在许多例子中，一个简单的设计好的图样能表达可能需要许多页纸才能解释清其复杂概念。作为可视化的核心，本书反映了在图形学，特别是计算机图形的能力方面不可动摇的信念。

本书是按实用和易用方式进行组织的。第一部分是重要的基础，它为后来的第二部分和第三部分的高级编程技术提供最基本的构造模块。

本书不想强调读者教条地去学。如果你是一个初学者或中等程度的程序员，你可以仔细地从头学到尾，定会从中获益；如果你已是中等以上的程序员，最好能一节节地跳读，选择你最感兴趣的内容。

第一部分介绍了建立可视化软件的编程技巧，共分七章。

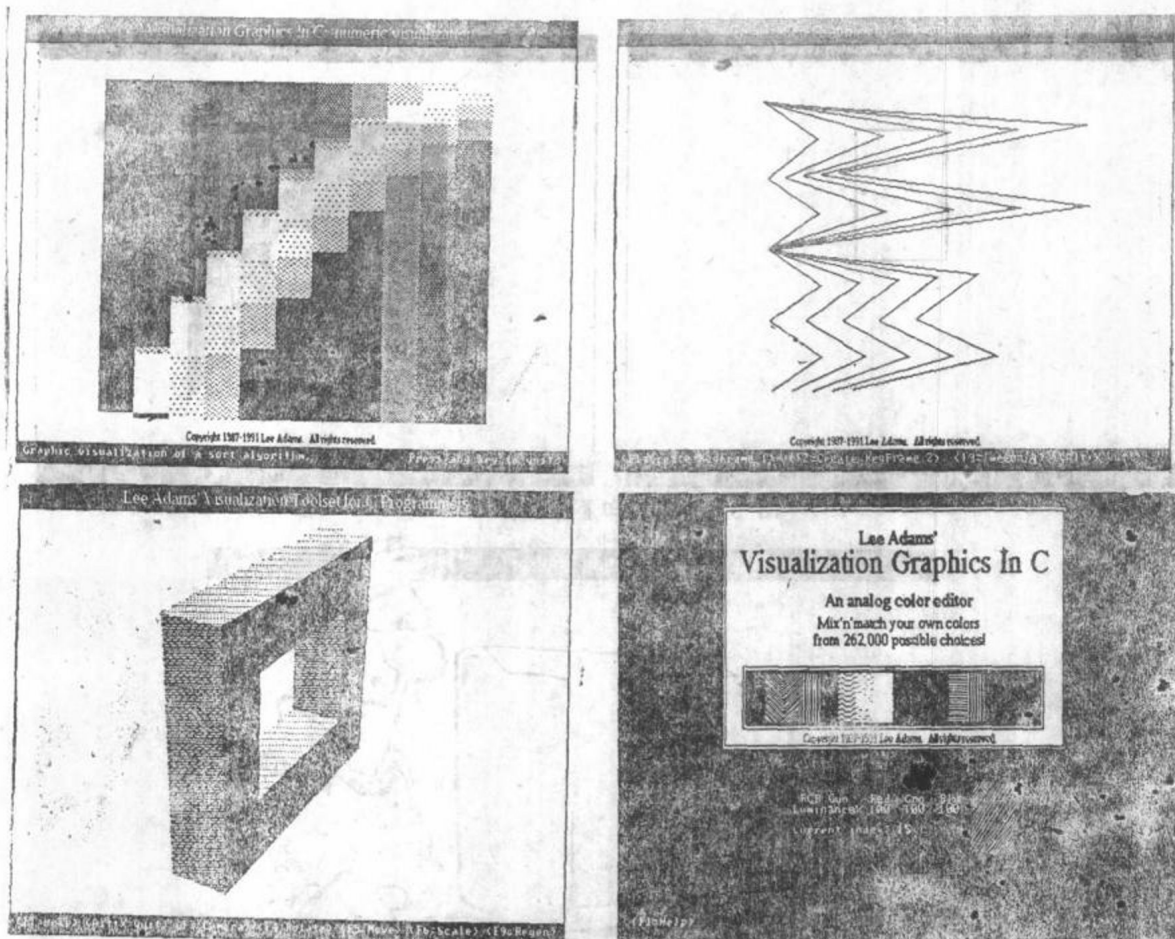


图 8 屏幕打印 I

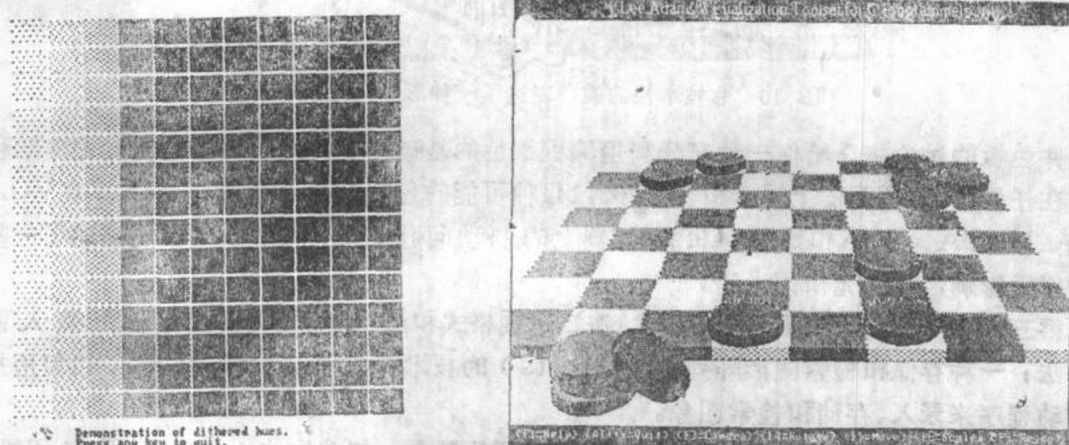


图 9 屏幕打印 II

第一章讨论了针对应用程序开发的实用技术，包括能使你的程序运行在不同图形适配器上的预处理伪指令的使用。你会看到如何更有效地使用Quick C和Turbo C编辑器和编译器；你也能学到如何用不同的语言来加一些屏幕信息，如何检索用户键入的命令行变元等。

第二章提供了关于色彩（hues）的实用技术。关于VGA如何处理多达262,000种颜色

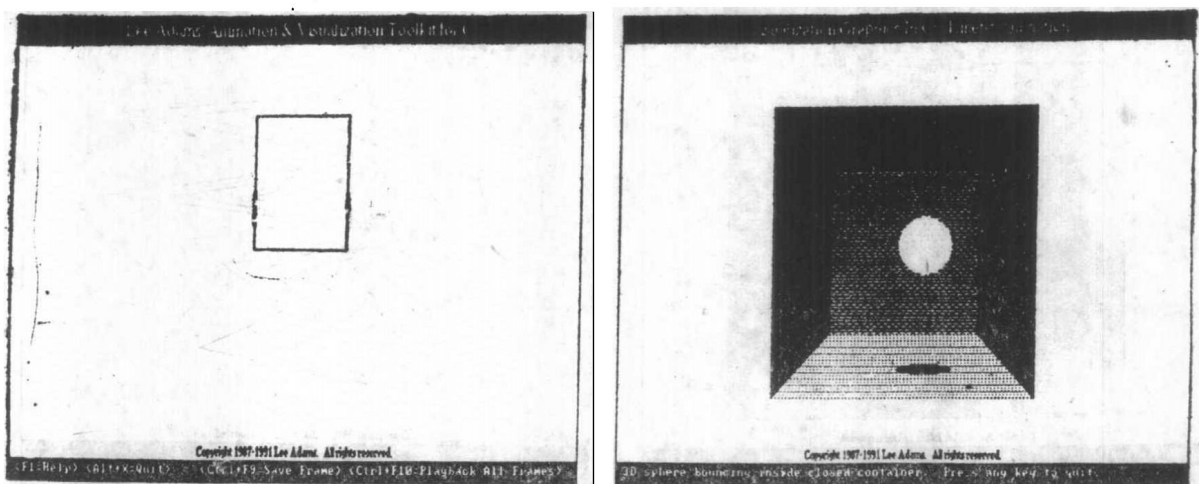


图 9 屏幕打印 I (续)

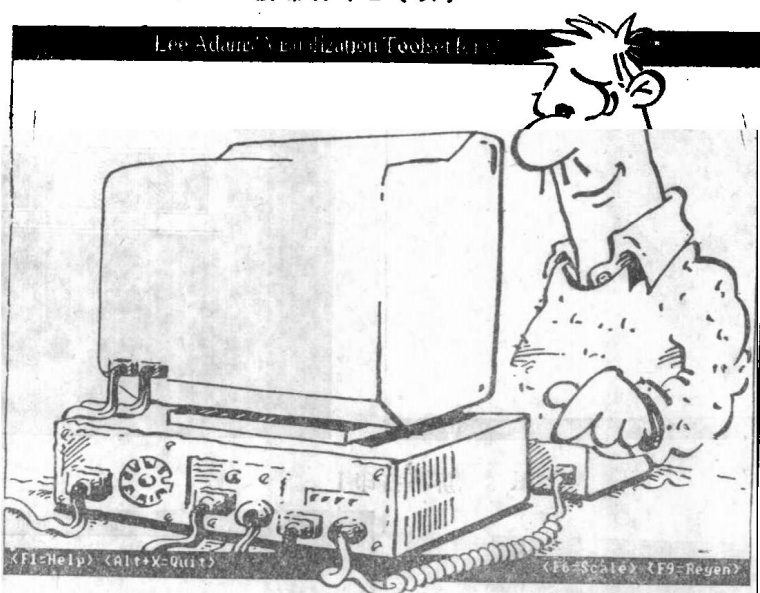


图 10 非技术性的插图表达了一种形象化的概念和思想。

的硬件色板的讨论将会给你一种充分利用编程颜色的基础知识。一个完整的交互式演示程序让你在任何VGA适配器上混合和匹配262,144种可能的色彩；第二个程序演示了在VGA、EGA、MCGA、CGA和Hercules适配器上的抖动问题(dithering)。还描述了一些特殊的视觉影响，如发光和闪光等。

第三章阐述了图象处理的实用技术：按二进制格式和流行的PCX格式存储和装入图象的方法；一种存储和检索图形阵列（即bitblts）的技术；此外还向读者显示了如何用扫描仪驱动程序来录入、存储和检索图象。

第四章讲述了基本的动画编程技术：一些重要的概念，如动作体（角色）、固定相机、移动相机、基于RAM的实时动画、基于磁盘的帧动画等也被引入和介绍；一些重要的代码段，如怎样精确控制帧速率，如何管理图形阵列动画，如何设置带前导、反导、单步和定格能力的交互式重放控制器等。

第五章描述了鼠标和数字化板的编程。你会学会和掌握每种定点设备的能力和弱点，而且会看到一组实用的鼠标例程。

第六章处理声音的编程问题。你会学到如何训练一个语音识别系统来处理语音命令，同时还介绍按一给定的音调和音长来生成声音的例程。

第七章讨论了前端界面(front_end)编程问题，这主要集中在菜单系统、控制板和行编辑器。一个演示程序显示了在一种运行于图形方式下的C语言程序里如何提供全特征的文本编辑器。另一个演示程序提供了一个可为你的图形程序所用的下拉式菜单系统外壳。

第二部分(第八章至第十三章)提供了与物体可视化有关材料，共包括三维实体造型和真实感图形生成、表面浓淡处理、平滑浓淡处理、隐藏面消除、程式动画和运动动画。

第八章介绍了与三维编程有关的基本概念。对边界表示法(B_rep)和构造实体几何法(CSG)都进行了描述。对二次三维图素和浓淡处理方法学也进行了讨论；对象凸凹纹理生成、表面反射等高级技术也进行了研究。

第九章是关于表面浓淡处理的有用材料，你可以在你的配有VGA、EGA、MCGA、CGA和Hercules图形适配器的PC机上从中学到实现完全的有明暗处理的三维实体模型的效果好效率高的方法。提供了你能用一个完整的交互式演示程序来构造复杂三维景物的三维实体造型和浓淡处理编辑器。编辑器提供了自动的后平面的消除、元文件的存储与装入以及有用的重生成函数功能。演示程序还提供了七种不同语言的屏幕信息。

第十章针对三维实体造型的平滑浓淡处理算法进行了探讨。马赫带及数字虚象也进行讨论。就一个显示如何在PC机上实现平滑浓淡处理的色调插值方法的详细算法也作了细致的论述。

第十一章研究了隐藏面消除问题，对目标(物体)空间和图象空间的考虑方法也进行了评价。一个源代码片段提供了由几个三维实体组成的景物中隐藏面消除的易于理解的工具。

第十二章与程式动画有关，描述了脚本(剧本)动画和动作体(角色)，同时表述了静态观点的固定相机技术、动态观点的移动相机技术。一个演示程序提供了能用来构造复杂动画序列的交互编辑器。程序还提供了一个带有实时前导、反导、单步和定格的交互式重放控制器。

第十三章探讨了运动动画的概念和技术；科学可视化原理、物理学定律等；对质量、动量、速率和能量守恒等也进行了讨论。同时还研究了碰撞检测、刚体和弹性，一个演示程序显示了如何管理一个球体从一个三维容器的壁上反弹的可视化过程。

第三部分涉及图象可视化的题目，包括图象处理、计算机视觉、图象的变形与中间帧生成cel动画。

第十四章讨论了通常用在图象处理中的象素操作和变换，研究了诸如延伸、反转、镜象移动等几何处理技术。也描述了边缘检测，图象比较，相片润色等也进行，还讨论了扫描图象的编辑和调色技术。对傅里叶分析、超级采样和过滤技术也作了介绍。用一个演示程序提供了做这些工作的计算机算法的可视化。

第十五章计算机视觉、讨论了边缘检测和模式识别以及软硬件的需求。

第十六章介绍了图象变形与中间帧生成技术，即描述如何利用关键帧来构造一系列对动画和插图有用的图象。对线性插值和样条插值也进行了讨论。一个完整的交互式演示程序提供了用来构造关键帧的图形编辑器。编辑器也能为你自动地建立所有的中间图象。

除正文外，本书还提供了一系列的附录。

附录一提供了一种并非废话的指南，它会告诉你如何准备演示程序并在你的系统上用

Quick C和Turbo C运行它们。

附录二提供了如何选择和使用第三家库的指南。

附录三显示出如何在C图形程序运行时捕捉错误。

附录四提供了使用扩充内存和扩展内存写图形程序的具体指导

附录五介绍了C++图形编程技术,估价了Zortech C++和Turbo C++的能力。

附录六研究了现实世界的高级可视化应用。对图形协处理器也进行了讨论。你还将学到如何处理NTSC格式的图象,以及计算机同科学仪器接口的技术。

附录七提供了工具包模块的全部源代码。这些工具包是演示程序中复杂图形特征的基础。

此外,还有术语汇编、中英关键字(词)索引、C语言介绍及商品化的3D产品等内容供读者使用。

尽管本书中的演示程序可以产生极为复杂的屏幕图象,但对你的软硬件配置要求并不高,你能在一种典型的PC机上使用通常的C编译器来开发你自己的可视化图形的潜力。

本书中的演示程序建立在与IBM PC兼容的使用Intel 8086,8808,80286,80386或80486微处理器的硬件平台上。你可以使用VGA、EGA、MCGA、CGA或Hercules适配器及其相应的显示器。另外你得准备一个硬盘及640KB的RAM。

你可以使用Quick C或Turbo C来编译、链接和运行演示程序。源代码也可以不经修改使用Microsoft C优化编译器5.10或Microsoft C个人开发系统PDS6.00来成功地编译。

程序清单中使用的是Quick C 2.00和Turbo C 2.0,你也可以使用这种版本或者更新的版本。注意,Quick C 1.00缺少2.00版中的一些图形函数;Turbo C 1.0没有图形库,1.5则缺乏2.0版中的许多高级特征。

有关源程序清单的问题可参阅有关内容,愿你能从后续章节以及附录中得到满意结果。

注,①G.Domik, The role of visualization in understanding data, New results and new trends in computer science, H.Maurer(Ed.)Springer-Verlag 1991, pp.91-107

②K.W.Brodie, "Scientific visualization"——Techniques and applications, Springer-Verlag 1992