经 典 原 版 书 库

计算机图形学原理及实践

C 语言描述

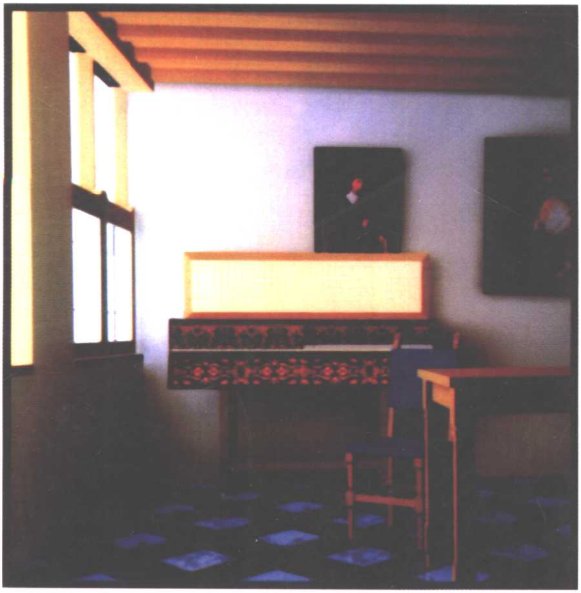
（英文版 · 第2版）

Computer Graphics

PRINCIPLES AND PRACTICE

Foley ◆ van Dam ◆ Feiner ◆ Hughes

SECOND EDITION in C



THE SYSTEMS PROGRAMMING SERIES

（美）James D. Foley 等 著

机械工业出版社
China Machine Press

Pearson Education
培生教育出版集团

经典原版书库

计算机图形学原理及实践

C语言描述

（英文版·第2版）

Computer Graphics

Principles and Practice Second Edition in C

（美）James D. Foley 等 著

机械工业出版社
China Machine Press

Pearson Education
培生教育出版集团

本书版权登记号：图字：01-2002-2188

图书在版编目（CIP）数据

计算机图形学原理及实践：C语言描述（英文版·第2版）/（美）福利（Foley, J.D.）等著. -北京：机械工业出版社，2002.6

（经典原版书库）

书名原文：Computer Graphics: Principles and Practice Second Edition in C

ISBN 7-111-10343-2

Ⅰ. 计…　Ⅱ. 福…　Ⅲ. 计算机图形学-英文　Ⅳ. TP391.41

中国版本图书馆CIP数据核字（2002）第035324号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）

责任编辑：华　章

北京昌平奔腾印刷厂印刷·新华书店北京发行所发行

2002年6月第1版第1次印刷

850mm × 1168mm 1/32 · 39.875印张

印数：0 001-3 000册

定价：88.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

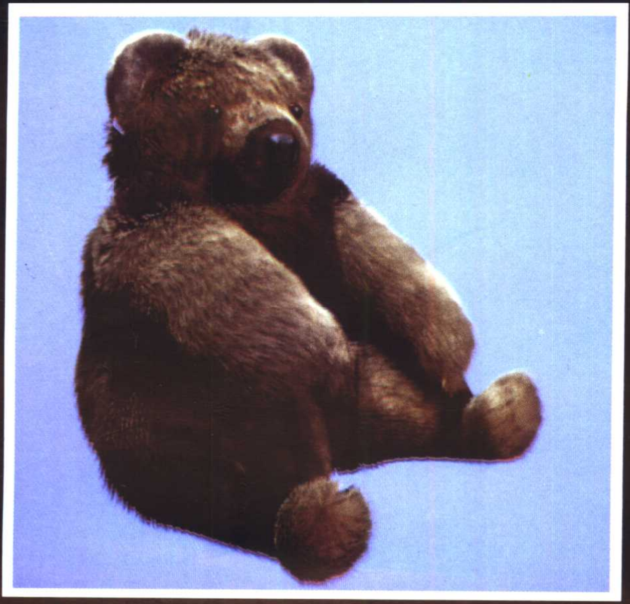


Plate A "Lilac the Bear," by J. Kajiya, T. Kay, and J. Snyder. (Produced at Caltech and IBM Corporation. Copyright © 1989 Caltech.)

Plate C "Voxel Garden." (Courtesy of Ned Greene, NYIT Computer Graphics Lab

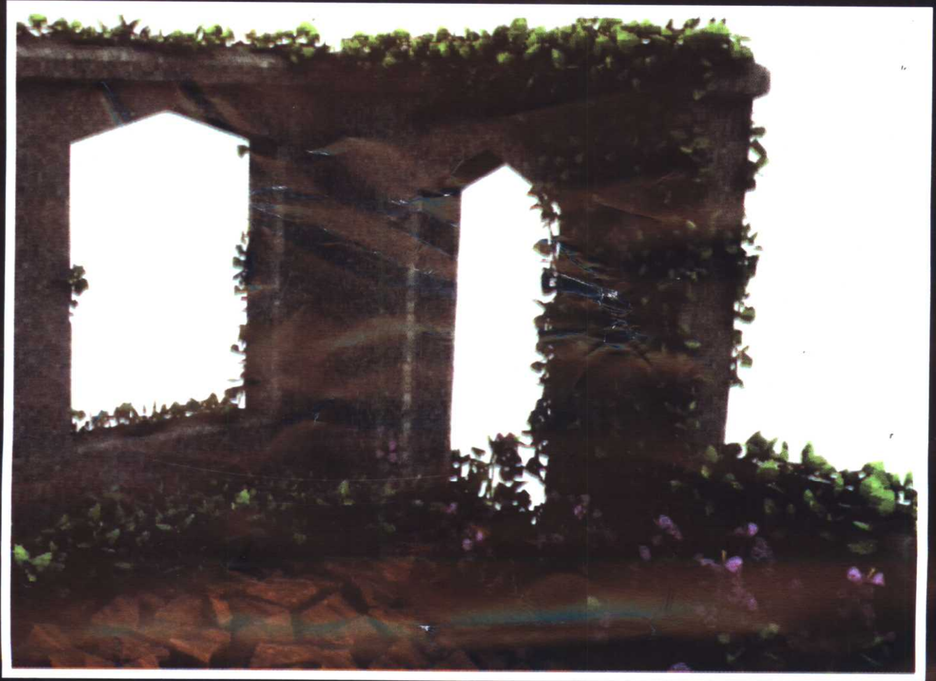
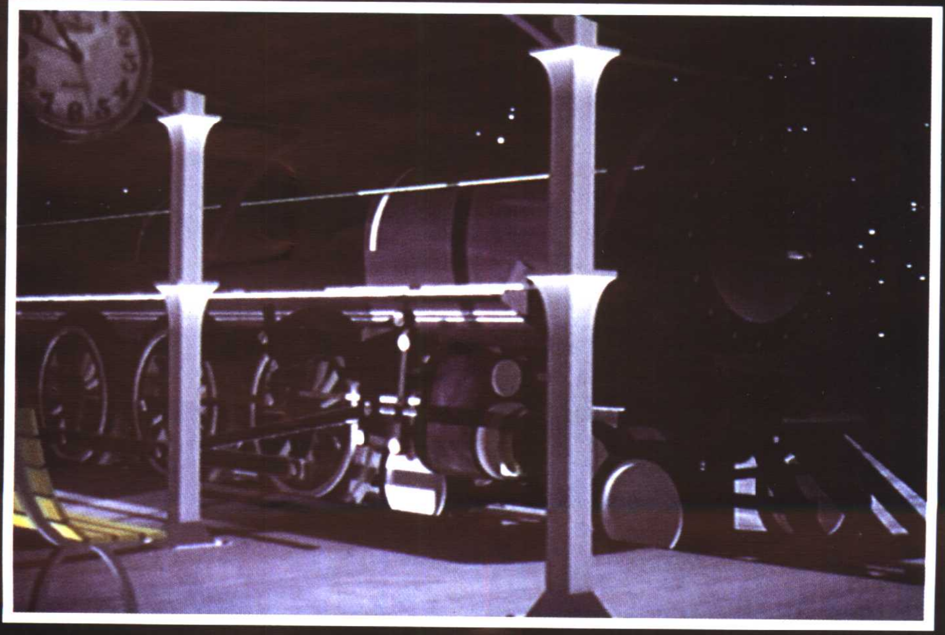


Plate B "Blessed State," by F. K. Musgrave. (Copyright © 1989 K. Musgrave and B. Mandelbrot.)

Plate D "Luxo Jr.," by J. Lasseter, W. Reeves, E. Ostby, and S. Leffler. (Copyright © 1986 Pixar.)

Plate E Ray-traced radiosity simulation, by K. Howie, B. Trumbore, and D.P. Greenberg, Program of Computer Graphics, Cornell University. (Copyright © 1989 Cornell University, Program of Computer Graphics.)

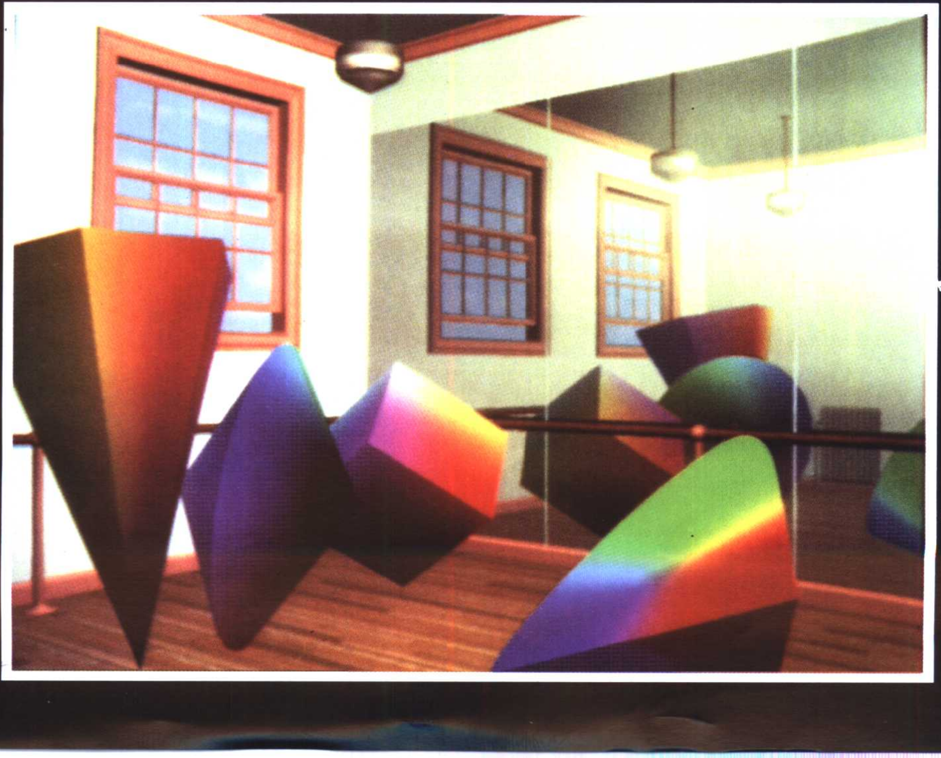
Plate G "Onlyville," by S. Snibbe, and D. Robbins. (Copyright © 1989 Brown University Computer Graphics Group.)



SALE
Clearance

Plate F "Red's Dream," by J. Lasseter, E. Ostby, W. Reeves, and H.B. Siegel. (Copyright © 1987 Pixar.)

Plate H "Color Choreography." Four color spaces are shown: HSV hexcone, the Ostwald double cone, the RGB cube, and the CIE color space. (Image by D. Laidlaw and B. Meier; Copyright © Laidlaw/Meier, Brown University Computer Graphics Group.)



出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall，Addison-Wesley，McGraw-Hill，Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum，Stroustrup，Kernighan，Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及庋藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：针对本科生的核心课程，剔抉外版菁华而成“国外经典教材”系列；对影印版的教材，则单独开辟出“经典原版书库”；定位在高级教程和专业参考的“计算机科学丛书”还将保持原来的风格，继续出版新的品种。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师们服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

“经典原版书库”是响应教育部提出的使用原版国外教材的号召，为国内高校的计算机教学度身订造的。在广泛地征求并听取丛书的“专家指导委员会”的意见后，我们最终选定了这30多种篇幅内容适度、讲解鞭辟入里的教材，其中的大部分已经被M.I.T.、Stanford、U.C. Berkley、C.M.U.等世界名牌大学采用。丛书不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

电子邮件：hzedu@hzbook.com
联系电话：(010)68995265
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

专家指导委员会

Foreword

The field of systems programming primarily grew out of the efforts of many programmers and managers whose creative energy went into producing practical, utilitarian systems programs needed by the rapidly growing computer industry. Programming was practiced as an art where each programmer invented his own solutions to problems with little guidance beyond that provided by his immediate associates. In 1968, the late Ascher Opler, then at IBM, recognized that it was necessary to bring programming knowledge together in a form that would be accessible to all systems programmers. Surveying the state of the art, he decided that enough useful material existed to justify a significant codification effort. On his recommendation, IBM decided to sponsor The Systems Programming Series as a long term project to collect, organize, and publish those principles and techniques that would have lasting value throughout the industry. Since 1968 eighteen titles have been published in the Series, of which six are currently in print.

The Series consists of an open-ended collection of text-reference books. The contents of each book represent the individual author's view of the subject area and do not necessarily reflect the views of the IBM Corporation. Each is organized for course use but is detailed enough for reference.

Representative topic areas already published, or that are contemplated to be covered by the Series, include: database systems, communication systems, graphics systems, expert systems, and programming process management. Other topic areas will be included as the systems programming discipline evolves and develops.

The Editorial Board

Preface

> Interactive graphics is a field whose time has come. Until recently it was an esoteric specialty involving expensive display hardware, substantial computer resources, and idiosyncratic software. In the last few years, however, it has benefited from the steady and sometimes even spectacular reduction in the hardware price/performance ratio (e.g., personal computers for home or office with their standard graphics terminals), and from the development of high-level, device-independent graphics packages that help make graphics programming rational and straightforward. Interactive graphics is now finally ready to fulfill its promise to provide us with pictorial communication and thus to become a major facilitator of man/machine interaction. (From preface, *Fundamentals of Interactive Computer Graphics,* James Foley and Andries van Dam, 1982)

This assertion that computer graphics had finally arrived was made before the revolution in computer culture sparked by Apple's Macintosh and the IBM PC and its clones. Now even preschool children are comfortable with interactive-graphics techniques, such as the desktop metaphor for window manipulation and menu and icon selection with a mouse. Graphics-based user interfaces have made productive users of neophytes, and the desk without its graphics computer is increasingly rare.

At the same time that interactive graphics has become common in user interfaces and visualization of data and objects, the rendering of 3D objects has become dramatically more realistic, as evidenced by the ubiquitous computer-generated commercials and movie special effects. Techniques that were experimental in the early eighties are now standard practice, and more remarkable "photorealistic" effects are around the corner. The simpler kinds of pseudorealism, which took hours of computer time per image in the early eighties, now are done routinely at animation rates (ten or more frames/second) on personal computers. Thus "real-time" vector displays in 1981 showed moving wire-frame objects made of tens of thousands of vectors without hidden-edge removal; in 1990 real-time raster displays can show not only the same kinds of line drawings but also moving objects composed of as many as one hundred thousand triangles rendered with Gouraud or Phong shading and specular highlights and with full hidden-surface removal. The highest-performance systems provide real-time texture mapping, antialiasing, atmospheric attenuation for fog and haze, and other advanced effects.

Graphics software standards have also advanced significantly since our first edition. The SIGGRAPH Core '79 package, on which the first edition's SGP package was based, has all but disappeared, along with direct-view storage tube and refresh vector displays. The much more powerful PHIGS package, supporting storage and editing of structure hierarchy, has become an official ANSI and ISO standard, and it is widely available for real-time geometric graphics in scientific and engineering applications, along with PHIGS+, which supports lighting, shading, curves, and surfaces. Official graphics standards complement lower-level, more efficient de facto standards, such as Apple's QuickDraw, X Window System's Xlib 2D integer raster graphics package, and Silicon Graphics' GL 3D library. Also widely available are implementations of Pixar's RenderMan interface for photorealistic rendering and PostScript interpreters for hardcopy page and screen image description. Better graphics software has been used to make dramatic improvements in the "look and feel"of user interfaces, and we may expect increasing use of 3D effects, both for aesthetic reasons and for providing new metaphors for organizing and presenting, and navigating through information.

Perhaps the most important new movement in graphics is the increasing concern for modeling objects, not just for creating their pictures. Furthermore, interest is growing in describing the time-varying geometry and behavior of 3D objects. Thus graphics is increasingly concerned with simulation, animation, and a "back to physics" movement in both modeling and rendering in order to create objects that look and behave as realistically as possible.

As the tools and capabilities available become more and more sophisticated and complex, we need to be able to apply them effectively. Rendering is no longer the bottleneck. Therefore researchers are beginning to apply artificial-intelligence techniques to assist in the design of object models, in motion planning, and in the layout of effective 2D and 3D graphical presentations.

Today the frontiers of graphics are moving very rapidly, and a text that sets out to be a standard reference work must periodically be updated and expanded. This book is almost a total rewrite of the *Fundamentals of Interactive Computer Graphics,* and although this second edition contains nearly double the original 623 pages, we remain painfully aware of how much material we have been forced to omit.

Major differences from the first edition include the following:

- The vector-graphics orientation is replaced by a raster orientation.
- The simple 2D floating-point graphics package (SGP) is replaced by two packages—SRGP and SPHIGS—that reflect the two major schools of interactive graphics programming. SRGP combines features of the QuickDraw and Xlib 2D integer raster graphics packages. SPHIGS, based on PHIGS, provides the fundamental features of a 3D floating-point package with hierarchical display lists. We explain how to do applications programming in each of these packages and show how to implement the basic clipping, scan-conversion, viewing, and display list traversal algorithms that underlie these systems.
- User-interface issues are discussed at considerable length, both for 2D desktop metaphors and for 3D interaction devices.
- Coverage of modeling is expanded to include NURB (nonuniform rational B-spline) curves and surfaces, a chapter on solid modeling, and a chapter on advanced modeling techniques, such as physically based modeling, procedural models, fractals, L-grammar systems, and particle systems.
- Increased coverage of rendering includes a detailed treatment of antialiasing and greatly

expanded chapters on visible-surface determination, illumination, and shading, including physically based illumination models, ray tracing, and radiosity.

- Material is added on advanced raster graphics architectures and algorithms, including clipping and scan-conversion of complex primitives and simple image-processing operations, such as compositing.
- A brief introduction to animation is added.

This text can be used by those without prior background in graphics and only some background in Pascal programming, basic data structures and algorithms, computer architecture, and simple linear algebra. An appendix reviews the necessary mathematical foundations. The book covers enough material for a full-year course, but is partitioned into groups to make selective coverage possible. The reader, therefore, can progress through a carefully designed sequence of units, starting with simple, generally applicable fundamentals and ending with more complex and specialized subjects.

Basic Group. Chapter 1 provides a historical perspective and some fundamental issues in hardware, software, and applications. Chapters 2 and 3 describe, respectively, the use and the implementation of SRGP, a simple 2D integer graphics package. Chapter 4 introduces graphics hardware, including some hints about how to use hardware in implementing the operations described in the preceding chapters. The next two chapters, 5 and 6, introduce the ideas of transformations in the plane and 3-space, representations by matrices. the use of homogeneous coordinates to unify linear and affine transformations, and the description of 3D views, including the transformations from arbitrary view volumes to canonical view volumes. Finally, Chapter 7 introduces SPHIGS, a 3D floating-point hierarchical graphics package that is a simplified version of the PHIGS standard, and describes its use in some basic modeling operations. Chapter 7 also discusses the advantages and disadvantages of the hierarchy available in PHIGS and the structure of applications that use this graphics package.

User Interface Group. Chapters 8-10 describe the current technology of interaction devices and then address the higher-level issues in user-interface design. Various popular user-interface paradigms are described and critiqued. In the final chapter user-interface software, such as window managers, interaction technique-libraries, and user-interface management systems, is addressed.

Model Definition Group. The first two modeling chapters, 11 and 12, describe the current technologies used in geometric modeling: the representation of curves and surfaces by parametric functions, especially cubic splines, and the representation of solids by various techniques, including boundary representations and CSG models. Chapter 13 introduces the human color-vision system, various color-description systems, and conversion from one to another. This chapter also briefly addresses rules for the effective use of color.

Image Synthesis Group. Chapter 14, the first in a four-chapter sequence, describes the quest for realism from the earliest vector drawings to state-of-the-art shaded graphics. The artifacts caused by aliasing are of crucial concern in raster graphics, and this chapter discusses their causes and cures in considerable detail by introducing the Fourier transform and convolution. Chapter 15 describes a variety of strategies for visible-surface determination in enough detail to allow the reader to implement some of the most important ones. Illumination and shading algorithms are covered in detail in Chapter 16. The early part of this chapter discusses algorithms most commonly found in current hardware, while the remainder treats texture, shadows, transparency, reflections, physically based illumination models, ray tracing, and radiosity methods. The last chapter in this group, Chapter 17, describes both image manipulations, such as scaling, shearing, and rotating pixmaps, and image storage techniques, including various image-compression schemes.

Advanced Techniques Group. The last four chapters give an overview of the current state of the art (a moving target, of course). Chapter 18 describes advanced graphics hardware used in high-end commercial and research machines; this chapter was contributed by Steven Molnar and Henry Fuchs, authorities on high-performance graphics architectures. Chapter 19 describes the complex raster algorithms used for such tasks as scan-converting arbitary conics, generating antialiased text, and implementing page-description languages, such as PostScript. The final two chapters survey some of the most important techniques in the fields of high-level modeling and computer animation.

The first two groups cover only elementary material and thus can be used for a basic course at the undergraduate level. A follow-on course can then use the more advanced chapters. Alternatively, instructors can assemble customized courses by picking chapters out of the various groups.

For example, a course designed to introduce students to primarily 2D graphics would include Chapters 1 and 2, simple scan conversion and clipping from Chapter 3, a technology overview with emphasis on raster architectures and interaction devices from Chapter 4, homogeneous mathematics from Chapter 5, and 3D viewing only from a "how to use it" point of view from Sections 6.1 to 6.3. The User Interface Group, Chapters 8-10, would be followed by selected introductory sections and simple algorithms from the Image Synthesis Group, Chapters 14, 15, and 16.

A one-course general overview of graphics would include Chapters 1 and 2, basic algorithms from Chapter 3, raster architectures and interaction devices from Chapter 4, Chapter 5, and most of Chapters 6 and 7 on viewing and SPHIGS. The second half of the course would include sections on modeling from Chapters 11 and 13, on image synthesis from Chapters 14, 15, and 16, and on advanced modeling from Chapter 20 to give breadth of coverage in these slightly more advanced areas.

A course emphasizing 3D modeling and rendering would start with Chapter 3 sections on scan converting, clipping of lines and polygons, and introducing antialiasing. The course would then progress to Chapters 5 and 6 on the basic mathematics of transformations and viewing, Chapter 13 on color, and then cover the key Chapters 14, 15, and 16 in the Image Synthesis Group. Coverage would be rounded off by selections in surface and solid modeling, Chapter 20 on advanced modeling, and Chapter 21 on animation from the Advanced Techniques Group.

Graphics Packages. The SRGP and SPHIGS graphics packages, designed by David Sklar, coauthor of the two chapters on these packages, are available from the publisher for the IBM PC (ISBN 0-201-54700-7), the Macintosh (ISBN 0-201-54701-5), and UNIX workstations running X11, as are many of the algorithms for scan conversion, clipping, and viewing (see page 1175).

Acknowledgments. This book could not have been produced without the dedicated work and the indulgence of many friends and colleagues. We acknowledge here our debt to those who have contributed significantly to one or more chapters; many others have helped by commenting on individual chapters, and we are grateful to them as well. We regret any inadvertent omissions. Katrina Avery and Lyn Dupré did a superb job of editing. Additional valuable editing on multiple versions of multiple chapters was provided by Debbie van Dam, Melissa Gold, and Clare Campbell. We are especially grateful to our production supervisor, Bette Aaronson, our art director, Joe Vetere, and our editor, Keith Wollman, not only for their great help in producing the book, but also for their patience and good humor under admittedly adverse circumstances—if we ever made a promised deadline during these frantic five years, we can't remember it!

Computer graphics has become too complex for even a team of four main authors and three guest authors to be expert in all areas. We relied on colleagues and students to amplify our knowledge, catch our mistakes and provide constructive criticism of form and content. We take full responsibility for any remaining sins of omission and commission. Detailed technical readings on one or more chapters were provided by John Airey, Kurt Akeley, Tom Banchoff, Brian Barsky, David Bates, Cliff Beshers, Gary Bishop, Peter Bono, Marvin Bunker, Bill Buxton, Edward Chang, Norman Chin, Michael F. Cohen, William Cowan, John Dennis, Tom Dewald, Scott Draves, Steve Drucker, Tom Duff, Richard Economy, David Ellsworth, Nick England, Jerry Farrell, Robin Forrest, Alain Fournier, Alan Freiden, Christina Gibbs, Melissa Gold, Mark Green, Cathleen Greenberg, Margaret Hagen, Griff Hamlin, Pat Hanrahan, John Heidema, Rob Jacob, Abid Kamran, Mike Kappel, Henry Kaufman, Karen Kendler, David Kurlander, David Laidlaw, Keith Lantz, Hsien-Che Lee, Aaron Marcus, Nelson Max, Deborah Mayhew, Barbara Meier, Gary Meyer, Jim Michener, Jakob Nielsen, Mark Nodine, Randy Pausch, Ari Requicha, David Rosenthal, David Salesin, Hanan Samet, James Sanford, James Sargent, Robin Schaufler, Robert Scheifler, John Schnizlein, Michael Shantzis, Ben Shneiderman, Ken Shoemake, Judith Schrier, John Sibert, Dave Simons, Jonathan Steinhart, Maureen Stone, Paul Strauss, Seth Tager, Peter Tanner, Brice Tebbs, Ben Trumbore, Yi Tso, Greg Turk, Jeff Vroom, Colin Ware, Gary Watkins, Chuck Weger, Kevin Weiler, Turner Whitted, George Wolberg, and Larry Wolff.

Several colleagues, including Jack Bresenham, Brian Barsky, Jerry Van Aken, Dilip Da Silva (who suggested the uniform midpoint treatment of Chapter 3) and Don Hatfield, not only read chapters closely but also provided detailed suggestions on algorithms.

Welcome word-processing relief was provided by Katrina Avery, Barbara Britten, Clare Campbell, Tina Cantor, Joyce Cavatoni, Louisa Hogan, Jenni Rodda, and Debbie van Dam. Drawings for Chapters 1–3 were ably created by Dan Robbins, Scott Snibbe, Tina Cantor, and Clare Campbell. Figure and image sequences created for this book were provided by Beth Cobb, David Kurlander, Allen Paeth, and George Wolberg (with assistance from Peter Karp). Plates II.21–37, showing a progression of rendering techniques, were designed and rendered at Pixar by Thomas Williams and H.B. Siegel, under the direction of M.W. Mantle, using Pixar's PhotoRealistic RenderMan software. Thanks to Industrial Light &

Magic for the use of their laser scanner to create Plates II.24–37, and to Norman Chin for computing vertex normals for Color Plates II.30–32. L. Lu and Carles Castellsagué wrote programs to make figures.

Jeff Vogel implemented the algorithms of Chapter 3, and he and Atul Butte verified the code in Chapters 2 and 7. David Sklar wrote the Mac and X11 implementations of SRGP and SPHIGS with help from Ron Balsys, Scott Boyajian, Atul Butte, Alex Contovounesios, and Scott Draves. Randy Pausch and his students ported the packages to the PC environment.

We have installed an automated electronic mail server to allow our readers to obtain machine-readable copies of many of the algorithms, suggest exercises, report errors in the text and in SRGP/SPHIGS, and obtain errata lists for the text and software. Send email to "graphtext @ cs.brown.edu" with a Subject line of "Help" to receive the current list of available services. (See page 1175 for information on how to order SRGP and SPHIGS.)

Preface to the C Edition

This is the C-language version of a book originally written with examples in Pascal. It includes all changes through the ninth printing of the Pascal second edition, as well as minor modifications to several algorithms, and all its Pascal code has been rewritten in ANSI C. The interfaces to the SRGP and SPHIGS graphics packages are now defined in C, rather than Pascal, and correspond to the new C implementations of these packages. (See page 1175 for information on obtaining the software.)

We wish to thank Norman Chin for converting the Pascal code of the second edition to C, proofreading it, and formatting it using the typographic conventions of the original. Thanks to Matt Ayers for careful proofing of Chapters 2, 3, and 7, and for useful suggestions about conversion problems.

Washington, D.C.	J.D.F.
Providence, R.I.	A.v.D.
New York, N.Y.	S.K.F.
Providence, R.I.	J.F.H.