THE
SYSTEMS
PROGRAMMING
SERIES

Computer Graphics: Principles and Practice

SECOND EDITION in C

Foley • van Dam • Feiner • Hughes

Sponsoring Editor: Peter S. Gordon
Production Supervisor: Bette J. Aaronson
Production Supervisor for the C edition: Juliet Silveri
Copy Editor: Lyn Dupré
Text Designer: Herb Caswell
Technical Art Consultant: Joseph K. Vetere
Illustrators: C&C Associates
Cover Designer: Marshall Henrichs
Manufacturing Manager: Roy Logan

This book is in the **Addison-Wesley Systems Programming Series**
Consulting editors: IBM Editorial Board

Library of Congress Cataloging-in-Publication Data

Computer graphics: principles and practice / James D. Foley . . . [et al.]. — 2nd ed. in C.
p. cm.
Includes bibliographical references and index.
ISBN 0-201-84840-6
1. Computer graphics. I. Foley, James D., 1942–
T385.C5735 1996
006.6'6—dc20 95-13631
CIP

Cover: "Dutch Interior," after Vermeer, by J. Wallace, M. Cohen, and D. Greenberg, Cornell University (Copyright © 1987 Cornell University, Program of Computer Graphics.)

Reprinted with corrections November 1992, November 1993, and July 1995.

3 4 5 6 7 8 9 10-DOC-99 98 97 96

To Marylou, Heather, Jenn, my parents, and my teachers

Jim

To Debbie, my father, my mother in memoriam, and
my children Elisa, Lori, and Katrin

Andy

To Jenni, my parents, and my teachers

Steve

To my family, my teacher Rob Kirby, and
my father in memoriam

John

And to all of our students.

THE SYSTEMS PROGRAMMING SERIES

Communications Architecture for Distributed Systems	R.J. Cypser
An Introduction to Database Systems, Volume I, Fifth Edition	C.J. Date
An Introduction to Database Systems, Volume II	C.J. Date
Computer Graphics: Principles and Practice, Second Edition	James D. Foley Andries van Dam Steven K. Feiner John F. Hughes
Structured Programming: Theory and Practice	Richard C. Linger Harlan D. Mills Bernard I. Witt
Conceptual Structures: Information Processing in Mind and Machines	John F. Sowa

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.

Contents

CHAPTER 7
OBJECT HIERARCHY AND SIMPLE PHIGS (SPHIGS) 285

CHAPTER 8
INPUT DEVICES, INTERACTION TECHNIQUES, AND INTERACTION TASKS 347

CHAPTER 9
DIALOGUE DESIGN 391

CHAPTER 10
USER INTERFACE SOFTWARE 435

CHAPTER 17
IMAGE MANIPULATION AND STORAGE 815

CHAPTER 18
ADVANCED RASTER GRAPHICS ARCHITECTURE 855

CHAPTER 19
ADVANCED GEOMETRIC AND RASTER ALGORITHMS 923