Computer Graphics

with OpenGL®

Computer Graphics

with OpenGL®

Fourth Edition

Donald Hearn

M. Pauline Baker

School of Informatics, Computer Science Department, and Pervasive Technology Labs at Indiana University
Indiana University—Purdue University Indianapolis

Warren R. Carithers

Department of Computer Science
Rochester Institute of Technology

Prentice Hall

Boston Columbus Indianapolis New York San Francisco Upper Saddle River
Amsterdam Cape Town Dubai London Madrid Milan Munich Paris Montréal Toronto
Delhi Mexico City São Paulo Sydney Hong Kong Seoul Singapore Taipei Tokyo

Editorial Director: *Marcia Horton*
Editor in Chief: *Michael Hirsch*
Executive Editor: *Tracy Dunkelberger*
Assistant Editor: *Melinda Haggerty*
Editorial Assistant: *Allison Michael*
Marketing Manager: *Yez Alayan*
Marketing Coordinator: *Kathryn Ferranti*
Vice President, Production: *Vince O'Brien*
Managing Editor: *Jeff Holcomb*
Senior Production Project Manager: *Marilyn Lloyd*
Senior Operations Supervisor: *Alan Fischer*
Operations Specialist: *Lisa McDowell*
Cover Director: *Linda Knowles*
Text Designer: *Janet Theurer/Theurer Briggs Design*
Permissions Specialist: *Dana Weightman*
Media Editor: *Dan Sandin*
Project Manager: *Martha Wetherill*
Full-Service Vendor: *MPS Limited, a Macmillan Company*
Printer/Binder: *Edwards Brothers*
Cover and Insert Printer: *Coral Graphics*

Cover Images: Created by Martin Wicke, Daniel Ritchie, Bryan M. Klingner, Sebastian Burke, Jonathan R. Shewchuk, and James F. O'Brien, University of California, Berkeley.

Credits and acknowledgments borrowed from other sources and reproduced, with permission, in this textbook appear on appropriate page within text.

Library of Congress Cataloging-in-Publication Data on file

10 9 8 7 6 5 4 3 2 1—EB—14 13 12 11 10

Prentice Hall
is an imprint of

PEARSON

www.pearsonhighered.com

ISBN 10: 0-13-605358-0
ISBN 13: 978-0-13-605358-3

TO MY FAMILY

Shelly, Catherine, and Max

Brief Contents

Contents

A Mathematics for Computer Graphics 731

B Graphics File Formats 767

Preface

Computer graphics remains one of the most exciting and rapidly growing areas of modern technology. Since the appearance of the first edition of this book, computer graphics has become a standard feature in applications software and computer systems in general. Computer-graphics methods are routinely applied in the design of most products, in training simulators, in the production of music videos and television commercials, in motion pictures, in data analysis, in scientific studies, in medical procedures, and in numerous other applications. A great variety of techniques and hardware devices are now in use or under development for these diverse application areas. Much of today's computer-graphics research continues to be concerned with improving the effectiveness, realism, and speed of picture generation. The difficulties involved in realistic rendering of complex materials such as hair, cloth, and fluids drive research in those areas, while image processing, animation, and surface representation continue to be popular areas of inquiry. The availability of advanced graphics hardware as a commodity item means that virtually any computer is capable of creating images of excellent quality, and the use of programmable graphics processing units is a research area of increasing interest and productivity.

New to the Fourth Edition

The material in this fourth edition has evolved from notes used in a variety of courses we have taught over the years, including introductory computer graphics, advanced computer graphics, scientific visualization, special topics, and project courses.

- New co-author, Warren Carithers, professor at Rochester Institute of Technology
- A new chapter introduces programmable shaders through the OpenGL Shading Language (GLSL)
- New material showcasing the evolution of OpenGL, a brief evaluation of changes in OpenGL 3.x and 4.x, and GPU architecture - past, present, and future.
- New material on the use of OpenGL in language other than C and C++, including Java and Python
- Implementation algorithms for graphics primitives and attributes moved into a single chapter
- Illumination models, texture mapping, and global illumination reorganized into separate, more tightly focused chapters
- Material on hierarchical modeling and animation moved earlier in the book
- Material on 3D object representations reorganized
- Material on 2D and 3D transformations and viewing reorganized
- 150 new exercises and new exercise set

Flexible Topic Order

Introductory computer graphics courses are taught using a wide variety of approaches and topic sets, and we have substantially reorganized the contents of many chapters and the chapter sequence in order to provide more flexibility in the way in which topics can be presented. Namely, implementation methods for primitives and attributes have been grouped into a single chapter and other large chapters that covered a wide variety of topics have been split into several smaller chapters, providing a more focused discussion of the material.

150 New Exercises and New Exercise Set

We have also revised or replaced over 150 of the exercises from the previous edition and created a new feature in the exercise section of each chapter labeled "In More Depth." These exercises provide, in many cases, a chapter-to-chapter continuity, allowing students to develop OpenGL programs that use advanced features in an incremental fashion.

Changes to OpenGL

With the availability of programmable GPUs, many graphics APIs (including OpenGL) are moving to the use of programmable shaders to provide more direct access to the power of the graphics hardware. Accordingly, material has been added to the book that introduces this very flexible approach to rendering, and a new appendix includes material on the evolution of OpenGL, the use of OpenGL from within languages other than C and C++, and an introduction to the capabilities of GPUs. OpenGL has evolved dramatically since the previous edition of this book was published. At that time, OpenGL had been in existence for just over a decade, and OpenGL 1.5 had just been released. While OpenGL had evolved to some degree in that time, it was still implemented using the original fixed-function pipeline model. Since then, there have been dramatic changes in the internal organization to allow it to make better use of current graphics hardware; in turn, these have resulted in significant changes to the OpenGL API.

During the preparation of this edition of the book, we debated whether or not to completely revise our introduction to OpenGL and use the new API. After much discussion and surveying instructors who teach the intro course, we decided to continue using the original interface in our discussions and examples in this edition, based on several factors:

- This book provides an introduction to OpenGL, and the original API is, arguably, easier for students new to graphics to learn.
- There is (and will be for the foreseeable future) a large base of existing OpenGL code that uses the original API.
- The most recent versions of OpenGL still support the original API in a compatibility mode.
- For several popular operating systems, the only available OpenGL implementations support only the original API.

About the Cover

The cover image shows stills from a simulation of a square plate fracturing when struck by a rigid projectile. The simulation was computed using a finite element code that dynamically restructures the mesh as the simulation progresses. The remeshing also adjusts the resolution of the simulation mesh, so that the complex

physical effects of fracture can be adequately resolved. The method is described in detail in Martin Wicke, Daniel Ritchie, Bryan M. Klingner, Sebastian Burke, Jonathan R. Shewchuk, and James F. O'Brien, "Dynamic Local Remeshing for Elastoplastic Simulation," in the proceedings of ACMSIGGRAPH 2010, article 49:1-11 (July 2010).

Programming Examples

More than twenty complete C++ programs are provided in this fourth edition, using the OpenGL, GLU, and GLUT libraries. These programs illustrate applications of basic picture-construction techniques, two-dimensional and three dimensional geometric transformations, two-dimensional and three-dimensional viewing methods, perspective projections, spline generation, fractal methods, interactive mouse input, picking operations, menu and submenu displays, and animation techniques. In addition, over one hundred C++/OpenGL program segments are given to demonstrate the implementation of computer-graphics algorithms for clipping, lighting effects, surface rendering, texture mapping, programmable shading, and many other computer-graphics methods.

Required Background

We assume no prior familiarity with computer graphics, but we do assume that the reader has some knowledge of computer programming and basic data structures, such as arrays, pointer lists, files, and record organizations. A variety of mathematical methods are used in computer-graphics algorithms, and these methods are discussed in some detail in Appendix A. Mathematical topics covered in Appendix A include techniques from analytic geometry, linear algebra, vector and tensor analysis, complex numbers, quaternions, basic calculus, and numerical analysis. This fourth edition can be used both as a text for students with no prior background in computer graphics and as a reference for graphics professionals. The emphasis is on the basic principles needed to design, use, and understand computer-graphics systems, along with numerous example programs to illustrate the methods and applications for each topic.

Suggested Course Outlines

One and/or Two Semester Course

For a one-semester course, a subset of topics dealing with either two-dimensional methods or a combination of two-dimensional and three-dimensional topics can be chosen, depending on the requirements of a particular course. A two-semester course sequence can cover the basic graphics concepts and algorithms in the first semester and advanced three-dimensional methods in the second.

At the undergraduate level, an introductory computer-graphics course can be organized using selected material from Chapters 2 through 10, and 17 through 20. Sections could be chosen from these chapters to cover two-dimensional or three-dimensional methods only (or a combination of the two), along with limited discussion of illumination and color. Other topics, such as fractal representations, spline curves, texture mapping, or depth-buffer methods, could be introduced in a first computer-graphics course.

For an introductory graduate or upper-level undergraduate course, more emphasis could be given to three-dimensional viewing, three-dimensional modeling, illumination models, and surface-rendering methods. In general, however, a two-semester sequence provides a better framework for adequately covering

the fundamentals of two-dimensional and three-dimensional computer-graphics methods, including spline representations, surface rendering, and ray tracing.

Special-topics courses, with an introductory computer-graphics prerequisite, can be offered in one or two areas, selected from visualization techniques, fractal geometry, spline methods, ray tracing, radiosity, and computer animation.

Self-Study

For the self-study reader, early chapters can be used to provide an understanding of graphics concepts, supplemented with selected topics from the later chapters.

Chapter-by-Chapter Synopsis

Chapter 1 illustrates the diversity of computer-graphics applications by taking a look at the many different kinds of pictures that people have generated with graphics software. In Chapter 2, we present the basic vocabulary of computer graphics, along with an introduction to the hardware and software components of graphics systems. Chapter 3 presents a detailed introduction to OpenGL, and a complete OpenGL example program. Chapters 4 through 6 introduce the fundamental methods for the representation and display of simple objects, and discuss methods for producing basic picture components such as polygons and circles, for setting the color, size, and other attributes of objects. Chapters 4 and 5 introduce these topics and discuss their use in OpenGL; Chapter 6 covers the underlying algorithms for drawing primitives and modifying attributes. Chapters 7 and 8 discuss the algorithms for performing geometric transformations such as rotation and scaling and viewing transformations in two-dimensional scenes; Chapters 9 and 10 do the same for three-dimensional scenes. Methods for the hierarchical modeling of complex systems are presented in Chapter 11. Computer-animation techniques are explored in Chapter 12. Methods for generating displays of complex objects, such as quadric surfaces, splines, and constructive solid geometry are discussed in Chapters 13, 14, and 15. In Chapter 16 we explore the various computer-graphics techniques for identifying the visible objects in a three dimensional scene. Illumination models and the methods for applying lighting conditions to a scene are examined in Chapter 17. Chapter 18 explores texturing and methods for representing surface detail. The various color models useful in computer graphics are discussed in Chapter 19, along with color-design considerations. Methods for interactive graphics input and for designing graphical user interfaces are given in Chapter 20. Chapter 21 discusses concepts related to global illumination. Programmable shaders are introduced in Chapter 22. Fractals, particle systems, and other algorithmic modeling techniques are explored in Chapter 23. Chapter 24 discusses visualization of data sets.

Instructor Resource Materials

The following protected instructor resource materials are available on the publisher's website at http://www.pearsonhighered.com/hearn. For username and password information, please contact your Pearson representative.

- Instructor solutions manual
- Downloadable source code
- Art and figure PowerPoint slides

Acknowledgments

Many people have contributed to this project in a variety of ways over the years. To the organizations and individuals who furnished pictures and other materials, we again express our appreciation. We also acknowledge the many helpful comments received from our students in various computer-graphics and visualization courses and seminars. We are indebted to all those who provided comments, reviews, suggestions for improving the material covered in this book, and other input, and we extend our apologies to anyone we may have failed to mention.

Our thanks to Ed Angel, Norman Badler, Phillip Barry, Brian Barsky, Hedley Bond, Bart Braden, Lara Burton, Robert Burton, Greg Chwelos, John Cross, Steve Cunningham, John DeCatrel, Victor Duvaneko, Gary Eerkes, Parris Egbert, Tony Faustini, Thomas Foley, Thomas Frank, Don Gillies, Andrew Glassner, Jack Goldfeather, Georges Grinstein, Eric Haines, Robert Herbst, Larry Hodges, Carol Hubbard, Eng-Kiat Koh, Mike Krogh, Michael Laszlo, Suzanne Lea, Michael May, Nelson Max, David McAllister, Jeffrey McConnell, Gary McDonald, C. L. Morgan, Greg Nielson, James Oliver, Lee-Hian Quek, Laurence Rainville, Paul Ross, David Salomon, Günther Schrack, Steven Shafer, Cliff Shaffer, Pete Shirley, Carol Smith, Stephanie Smullen, Jeff Spears, William Taffe, Wai Wan Tsang, Spencer Thomas, Sam Uselton, David Wen, Bill Wicker, Andrew Woo, Angelo Yfantis, Marek Zaremba, and Michael Zyda.

Special thanks go to those who contributed additional materials for this edition of the book, Rosario Leonardi (PERCRO Scuola Superiore Sant'Anna), Paul Nagin (Chimborazo Publishing, Inc.), James O'Brien (University of California, Berkeley), Emanuele Ruffaldi (PERCRO Scuola Superiore Sant'Anna), and Graham Sellers (Advanced Micro Devices, Inc.).

Our thanks also go to our reviewers, Emmanuel Agu (Worcester Polytechnic Institute), Ye Duan (University of Missouri, Columbia), John Hart (University of Illinois), Jong Kwan Lee (Bowling Green State University), Stephen Mann (University of Waterloo), Timothy Newman (University of Alabama, Huntsville), Amar Raheja (California State Polytechnic Institute, Pomona), Adrian Rusu (Rowan University), Jergen Schulze (University of California, San Diego), Soon Tee Teoh (San Jose State University), Iren Valova (University of Massachusettes, Dartmouth), Stephen Wismath (University of Lethbridge), and Dana Wortman (University of Colorado, Colorado Springs).

Finally, to our editors and production staff, Tracy Dunkelberger, Melinda Haggerty, Marilyn Lloyd, and Martha Wetherill, we extend our sincere appreciation for their help, suggestions, encouragement, careful attention to detail, and, above all, their patience during the preparation of this fourth edition.

About the Authors

Donald Hearn joined the Computer Science faculty at the University of Illinois at Urbana-Champaign in 1985. Dr. Hearn has taught a wide range of courses in computer graphics, scientific visualization, computational science, mathematics, and applied science. Also, he has directed numerous research projects and published a variety of technical articles in these areas.

M. Pauline Baker is on the faculty of the School of Informatics at Indiana University-Purdue University Indianapolis (IUPUI), where she is director of the Media Arts and Science program. She also directs the Visualization and Interactive Spaces Lab, part of the Pervasive Technology Institute at Indiana University. Before moving to Indiana, Prof. Baker was director of Visualization and Virtual Environments at the National Center for Supercomputing Applications (NCSA) at the University of Illinois. Prof. Baker holds a BS degree in Psychology (Cornell University), an MS degree in Education (Syracuse University), and a PhD in Computer Science (University of Illinois).

Warren R. Carithers joined the faculty of the Department of Computer Science at the Rochester Institute of Technology in 1981. In addition to teaching many of the department's courses in computer graphics, Professor Carithers develops and teaches courses in a wide range of other areas including operating systems, computer architecture and organization, systems software, programming language design, and security.