教 育 部 高 等 教 育 司 推 荐
国外优秀信息科学与技术系列教学用书

交互式计算机图形学

——自顶向下方法与 OpenGL 应用

(第三版　影印版)

INTERACTIVE COMPUTER GRAPHICS

A Top-Down Approach with OpenGL

(Third Edition)

■ Edward Angel

PEARSON
Addison Wesley

高等教育出版社
Higher Education Press

教 育 部 高 等 教 育 司 推 荐
国外优秀信息科学与技术系列教学用书

交互式计算机图形学
——自顶向下方法与 OpenGL 应用

（第三版　影印版）

INTERACTIVE COMPUTER GRAPHICS

A Top-Down Approach with OpenGL

(Third Edition)

Edward Angel

PEARSON
Addison Wesley

高 等 教 育 出 版 社

图字：01-2003-5098 号

Interactive Computer Graphics: A Top-Down Approach with OpenGL, Third Edition
Edward Angel

原版 ISBN: 0-201-77343-0

图书在版编目（C I P）数据

交互式计算机图形学：自顶向下方法与OpenGL应用＝Interactive Computer Graphics：A Top-Down Approach with OpenGL：第3版／（美）安杰尔（Angel,E.）著．—影印本．—北京：高等教育出版社，2003.10
ISBN 7－04－013724－0

Ⅰ.交…　Ⅱ.安…　Ⅲ.图形软件，OpenGL—高等学校—教材—英文　Ⅳ.TP391.41

中国版本图书馆 CIP 数据核字（2003）第088087号

出版发行	高等教育出版社	**购书热线**	010－64054588
社　　址	北京市西城区德外大街 4 号	**免费咨询**	800－810－0598
邮政编码	100011	**网　　址**	http://www.hep.edu.cn
传　　真	010－64014048		http://www.hep.com.cn
经　　销	新华书店北京发行所		
印　　刷	北京外文印刷厂		
开　　本	787×1092　1/16	**版　　次**	2003 年 10 月第 1 版
印　　张	47.25	**印　　次**	2003 年 10 月第 1 次印刷
字　　数	900 000	**定　　价**	54.00 元(附光盘)
插　　页	9		

前　　言

20 世纪末，以计算机和通信技术为代表的信息科学和技术对世界经济、科技、军事、教育和文化等产生了深刻影响。信息科学技术的迅速普及和应用，带动了世界范围信息产业的蓬勃发展，为许多国家带来了丰厚的回报。

进入 21 世纪，尤其随着我国加入 WTO，信息产业的国际竞争将更加激烈。我国信息产业虽然在 20 世纪末取得了迅猛发展，但与发达国家相比，甚至与印度、爱尔兰等国家相比，还有很大差距。国家信息化的发展速度和信息产业的国际竞争能力，最终都将取决于信息科学技术人才的质量和数量。引进国外信息科学和技术优秀教材，在有条件的学校推动开展英语授课或双语教学，是教育部为加快培养大批高质量的信息技术人才采取的一项重要举措。

为此，教育部要求由高等教育出版社首先开展信息科学和技术教材的引进试点工作。同时提出了两点要求，一是要高水平，二是要低价格。在高等教育出版社和信息科学技术引进教材专家组的努力下，经过比较短的时间，第一批引进的 20 多种教材已经陆续出版。这套教材出版后受到了广泛的好评，其中有不少是世界信息科学技术领域著名专家、教授的经典之作和反映信息科学技术最新进展的优秀作品，代表了目前世界信息科学技术教育的一流水平，而且价格也是最优惠的，与国内同类自编教材相当。

这项教材引进工作是在教育部高等教育司和高教社的共同组织下，由国内信息科学技术领域的专家、教授广泛参与，在对大量国外教材进行多次遴选的基础上，参考了国内和国外著名大学相关专业的课程设置进行系统引进的。其中，John Wiley 公司出版的贝尔实验室信息科学研究中心副总裁 Silberschatz 教授的经典著作《操作系统概念》，是我们经过反复谈判，做了很多努力才得以引进的。William Stallings 先生曾编写了在美国深受欢迎的信息科学技术系列教材，其中有多种教材获得过美国教材和学术著作者协会颁发的计算机科学与工程教材奖，这批引进教材中就有他的两本著作。留美中国学者 Jiawei Han 先生的《数据挖掘》是该领域中具有里程碑意义的著作。由达特茅斯学院 Thomas Cormen 和麻省理工学院、哥伦比亚大学的几

位学者共同编著的经典著作《算法导论》，在经历了 11 年的锤炼之后于 2001 年出版了第二版。目前任教于美国 Massachusetts 大学的 James Kurose 教授，曾在美国三所高校先后 10 次获得杰出教师或杰出教学奖，由他主编的《计算机网络》出版后，以其体系新颖、内容先进而倍受欢迎。在努力降低引进教材售价方面，高等教育出版社做了大量和细致的工作。这套引进的教材体现了权威性、系统性、先进性和经济性等特点。

教育部也希望国内和国外的出版商积极参与此项工作，共同促进中国信息技术教育和信息产业的发展。我们在与外商的谈判工作中，不仅要坚定不移地引进国外最优秀的教材，而且还要千方百计地将版权转让费降下来，要让引进教材的价格与国内自编教材相当，让广大教师和学生负担得起。中国的教育市场巨大，外国出版公司和国内出版社要通过扩大发行数量取得效益。

在引进教材的同时，我们还应做好消化吸收，注意学习国外先进的教学思想和教学方法，提高自编教材的水平，使我们的教学和教材在内容体系上，在理论与实践的结合上，在培养学生的动手能力上能有较大的突破和创新。

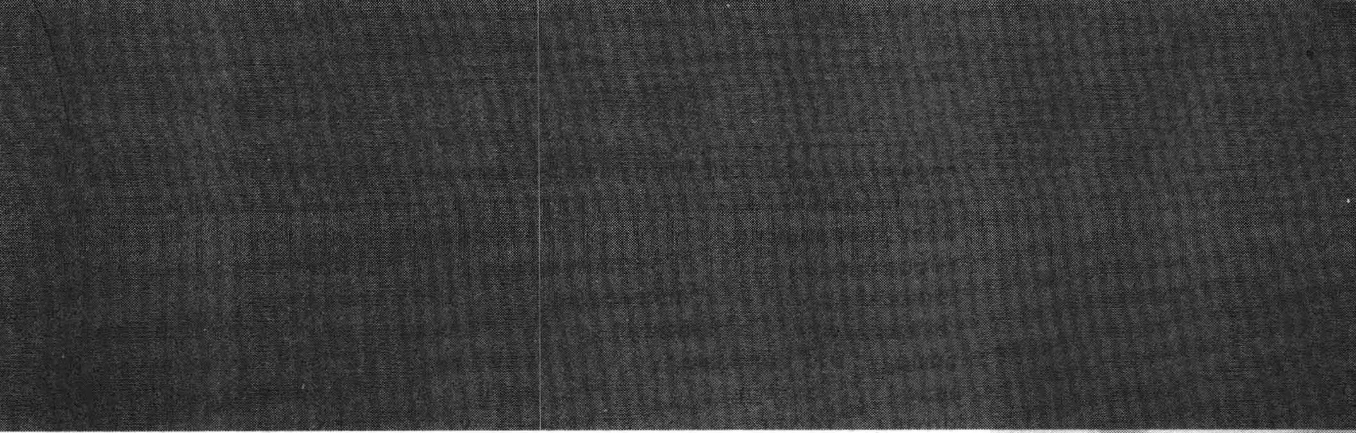
目前，教育部正在全国 35 所高校推动示范性软件学院的建设和实施，这也是加快培养信息科学技术人才的重要举措之一。示范性软件学院要立足于培养具有国际竞争力的实用性软件人才，与国外知名高校或著名企业合作办学，以国内外著名 IT 企业为实践教学基地，聘请国内外知名教授和软件专家授课，还要率先使用引进教材开展教学。

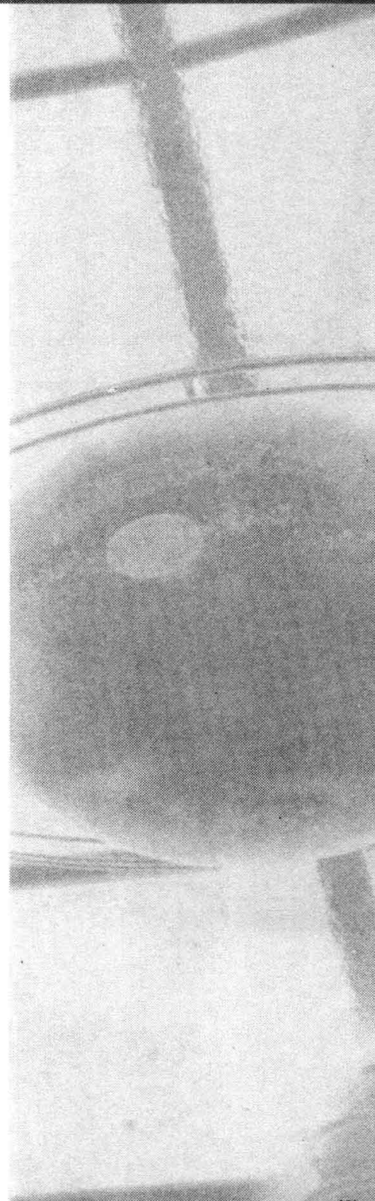
我们希望通过这些举措，能在较短的时间，为我国培养一大批高质量的信息技术人才，提高我国软件人才的国际竞争力，促进我国信息产业的快速发展，加快推动国家信息化进程，进而带动整个国民经济的跨越式发展。

教育部高等教育司

二〇〇二年三月

To Rose Mary





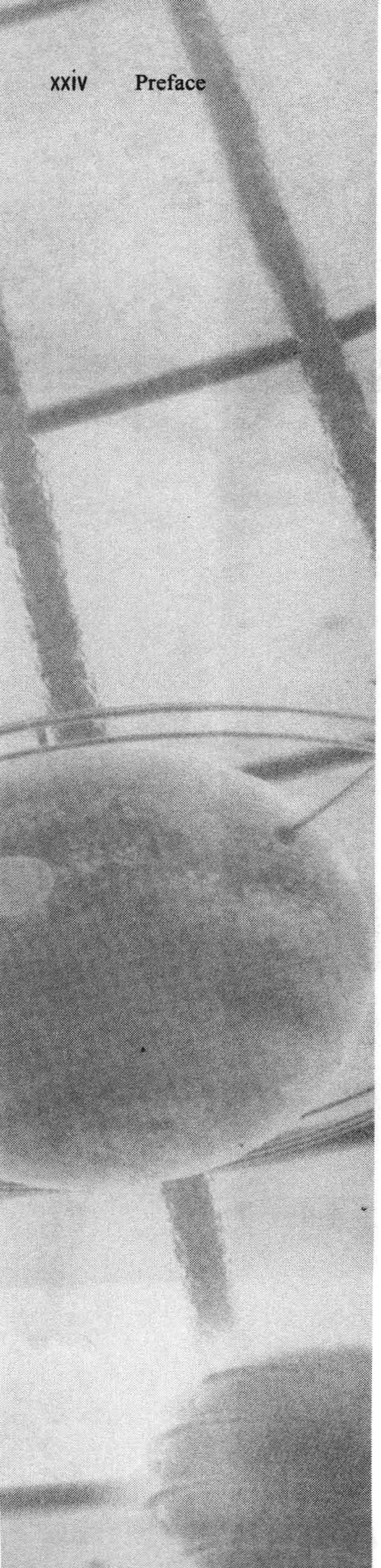
Preface

This book is an introduction to computer graphics, with an emphasis on applications programming. In the first edition, which was published in 1997, I noted that in the seven years since my previous graphics text, the field had experienced enormous growth—a rate of growth that exceeded most people's expectations, including my own. In the five years since, we have seen even more changes. Feature-length computer-animated movies have become commercial and artistic successes. The use of computer effects in movies is standard and it is often almost impossible to distinguish live action from computer-generated effects. Recent hardware has blurred the distinction between computers and game boxes. The explosion of interest in graphical applications over the Internet has continued.

Not only have graphics capabilities increased but costs have been reduced for both high- and low-end workstations. Within the last few years the cost of a graphics system that can generate over ten million three-dimensional polygons per second with lighting and texture mapping has dropped from over $100,000 to a few thousand dollars. Even computers that cost less than $1000 can support basic three-dimensional graphics applications. The availability of special-purpose graphics boards for personal computers has been especially significant. These boards provide workstation performance starting at about $100. On the software side, OpenGL remains the standard programmer's interface both for writing application programs and for developing high-level products in the scientific community.

A Top-Down Approach

These recent advances and the success of the first two editions of this text have reinforced my belief in a top-down, programming-oriented approach to introductory computer graphics. Although many computer science and engineering departments now support more than one course in the subject, most students will take only a single course. Such a course is placed in the curriculum after students have already studied programming, data structures, algorithms, software



engineering, and basic mathematics. A class in computer graphics allows the instructor to build on these topics in a way that can be both informative and fun. I want these students to be programming three-dimensional applications as soon as possible. Low-level algorithms, such as those that draw lines or fill polygons, can be dealt with later, after students are creating graphics.

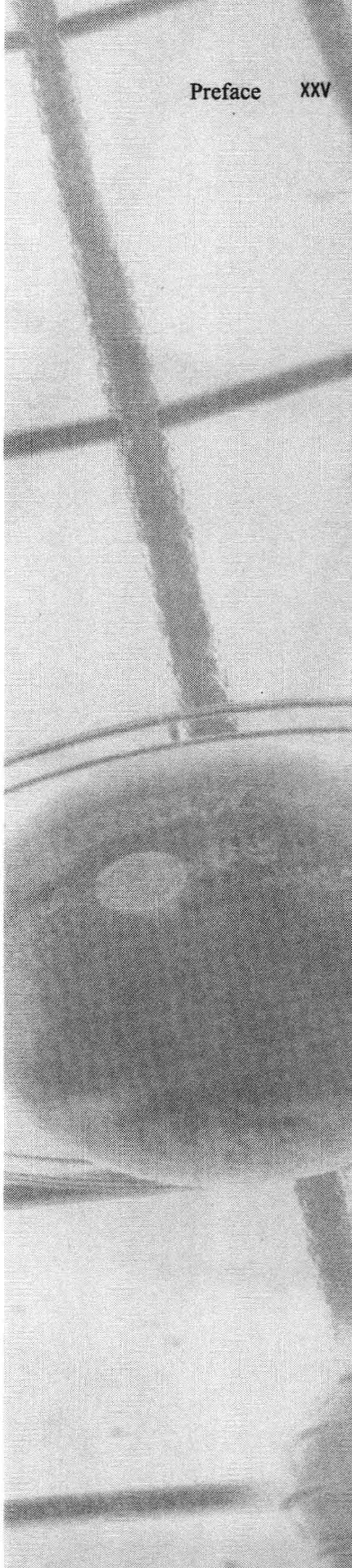
John Kemeny, a pioneer in computer education, used a familiar automobile analogy: You don't have to know what's under the hood to be literate, but unless you know how to program, you'll be sitting in the back seat instead of driving. That same analogy applies to the way we teach computer graphics. One approach—the algorithmic approach—is to teach everything about what makes a car function: the engine, the transmission, the combustion process. A second approach—the survey approach—is to hire a chauffeur, to sit back, and to see the world as a spectator. A third approach—the programming approach that I have adopted here—is to teach you how to drive and how to take yourself wherever you want to go. As the old auto-rental commercial used to say, "Let us put *you* in the driver's seat."

Programming with OpenGL and C

When I began teaching computer graphics 20 years ago, the greatest impediment to implementing a programming-oriented course, and to writing a textbook for that course, was the lack of a widely accepted graphics library or application programmer's interface (API). Difficulties included high cost, limited availability, lack of generality, and high complexity. The development of OpenGL resolved most of the difficulties many of us had experienced with other APIs (such as GKS and PHIGS), and with the alternative of using home-brewed software. OpenGL today is supported by most workstation suppliers and is available for most platforms through third-party vendors. It is bundled with all recent versions of Microsoft Windows and with the Apple Macintosh operating system. There is also an OpenGL API called Mesa that can be compiled for most systems and is included with most Linux distributions.

A graphics class teaches far more than the use of a particular API, but a good API makes it easier to teach key graphics topics, such as three-dimensional graphics, shading, client–server graphics, modeling, and implementation algorithms. OpenGL's extensive capabilities and well-defined architecture lead to a stronger foundation for teaching both theoretical and practical aspects of the field, and for teaching important new capabilities, such as texture mapping and compositing, that, until recently, were not supported in any API.

I switched my classes to OpenGL about 8 years ago, and the results astounded me. By the middle of the semester, *every* student was able to write a moderately complex three-dimensional program that required understanding of three-dimensional viewing and event-driven input. In 15 years of teaching computer graphics, I had never come even close to this result. That class led me to rewrite my previous book from scratch.



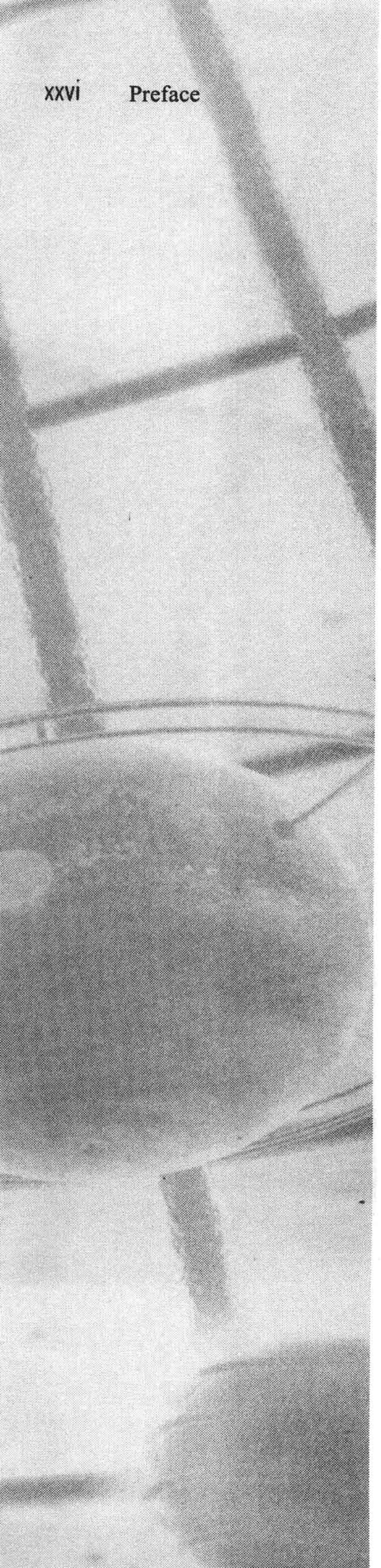
This is a textbook on computer graphics; it is not an OpenGL manual. Consequently, I do not cover all aspects of the OpenGL API, but rather explain only what is necessary for mastering this book's contents. I present OpenGL at a level that should permit users of other APIs to have little difficulty with the material. For students who want more detail on OpenGL, my recent book, *OpenGL: A Primer* (Addison-Wesley, 2002), should be a valuable supplement.

In this edition, I use both C and C++, with C as the dominant language. There are two reasons for this decision. First, OpenGL is not object-oriented, so using C++ or Java would not add significantly to the basic presentation, unless I were to insert an object-oriented geometric library between OpenGL and the user. I have not taken this step, despite its appealing features, because it would detract from the graphics and would make the book less accessible to students who are good programmers, but who are unfamiliar with object-oriented languages. Second, my experience has been that object-oriented approaches shield the user from what is going on inside (as they should), whereas, in an introduction to computer graphics, I want readers to be aware of what is happening at the lowest levels. Although the use of computer graphics is a wonderful way to introduce students to object-oriented programming, in my view, an object-oriented approach is not the most effective way to teach graphics to computer science and engineering students. The exception to this view is when I introduce scene graphs that are object oriented and benefit from the use of C++. My undergraduate students use C++ in their beginning courses, but have no difficulty in using the code in this book with either C or C++.

Within the computer graphics community, there has been much discussion of the future of OpenGL and whether it will be replaced by DirectX. Among computer graphics educators, however, there is little doubt that OpenGL will continue to be the API of choice for their classes. Although DirectX is the standard for game development, in the opinion of many of us, it is not well suited for teaching computer graphics or for users who want to develop their own applications. In addition to being specific to Windows platforms, DirectX requires far more code for basic portable applications. At the higher levels, OpenGL and DirectX support similar functionality in similar ways. Hence, users of this book should be able to move to DirectX with little difficulty when required.

Intended Audience

This book is suitable for advanced undergraduates and first-year graduate students in computer science and engineering and for students in other disciplines who have good programming skills. The book also will be useful to many professionals. I have taught approximately 100 short courses for professionals; my experiences with those students have had a great influence on what I have chosen to include in the book.



Prerequisites for using this text are good programming skills in C or C++, an understanding of basic data structures (linked list, trees), and a rudimentary knowledge of linear algebra and trigonometry. I have found that the mathematical backgrounds of computer science students, whether of undergraduates or of graduates, vary considerably. Hence, I have integrated much of the linear algebra and geometry that is required for fundamental computer graphics into the text. I have also summarized this material in two appendices.

Organization of the Book

The book is organized as follows. Chapter 1 overviews the field and introduces image formation by optical devices; thus, we start with three-dimensional concepts immediately. Chapter 2 introduces programming using OpenGL. Although the example program that we develop—each chapter has one or more complete programming examples—is two-dimensional, it is embedded in a three-dimensional setting. Chapter 3 discusses interactive graphics in a modern client–server setting and develops event-driven graphics programs. Chapters 4 and 5 concentrate on three-dimensional concepts; Chapter 4 is concerned with defining and manipulating three-dimensional objects, whereas Chapter 5 is concerned with viewing them. Chapter 6 introduces light–material interactions and shading. These chapters should be covered in order, and can be done in about 10 weeks of a 15-week semester.

The next seven chapters can be read in almost any order. All seven are somewhat open ended, and can be covered at a survey level, or individual topics can be pursued in depth. Chapter 7 introduces many of the new discrete capabilities that are now supported in graphics hardware and by OpenGL. All these techniques involve working with various buffers. It concludes with a short discussion of aliasing problems in computer graphics. Chapter 8 surveys implementation. It gives one or two major algorithms for each of the basic steps in the viewing pipeline. Chapter 9 contains a number of topics that fit loosely under the heading of hierarchical and object-oriented graphics. The topics range from building models that encapsulate the relationships between the parts of a model to high-level approaches to graphics over the Internet. It includes a simple scene graph API.

Curves and surfaces are discussed in Chapter 10. Chapter 11 introduces procedural methods as a way to escape the limits of building geometric models with polygons. It includes material on language-based models, fractals, and particle systems. Chapter 12 provides an introduction to scientific visualization. Not only is this area one of great current interest, it gives us a chance to demonstrate almost all the techniques developed in previous chapters. Chapter 13 is new and surveys alternate approaches to rendering. It includes expanded discussions of ray tracing and radiosity and an introduction to image-based rendering and parallel rendering.

Programs, primarily from the first part of the book, are included in Appendix A. They are also available electronically (see the Supplements section).

Changes from the Second Edition

The reaction of readers of the first and second editions of this book was overwhelmingly positive, especially to the use of OpenGL and the top-down approach. But there are always improvements to be made and recent advances to be added. This third edition addresses these issues.

I have made three types of changes from the second edition. First, I have enhanced the discussion of the more mathematical topics. Second, I have added material of current interest, such as scene graphs and alternate rendering methods. Third, I have included more OpenGL examples. These changes led to a reorganization of the second half of the book. These changes should have only minor effects on the teaching of a one semester introductory course, but should make it easier for instructors to use the book for a two-semester class.

Chapters 1 and 2 contain minor updates. In Chapter 3, I added material on the use of the exclusive-or writing mode so that readers would be able to develop applications that used rubberbanding. I also added a section on picking using selection mode.

Some of Chapter 4 has been rewritten to make the mathematical topics clearer. I also added a section on quaternions at the end of the chapter that fits well with the expanded treatment of smooth rotations in this edition.

Chapter 5 is largely unchanged.

Chapter 6 now focuses on pipeline rendering and the material on global rendering has been moved to Chapter 13.

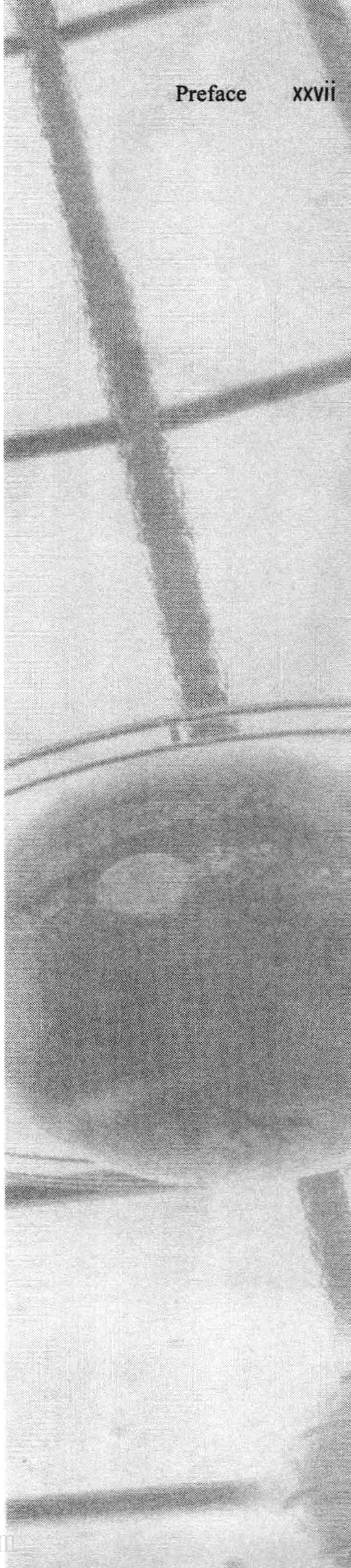
The information in Chapter 7 has been moved forward (from Chapter 9) to reflect the growing importance of mapping and the use of discrete buffers. There is more detail on using OpenGL's texture mapping functions and many additional code examples.

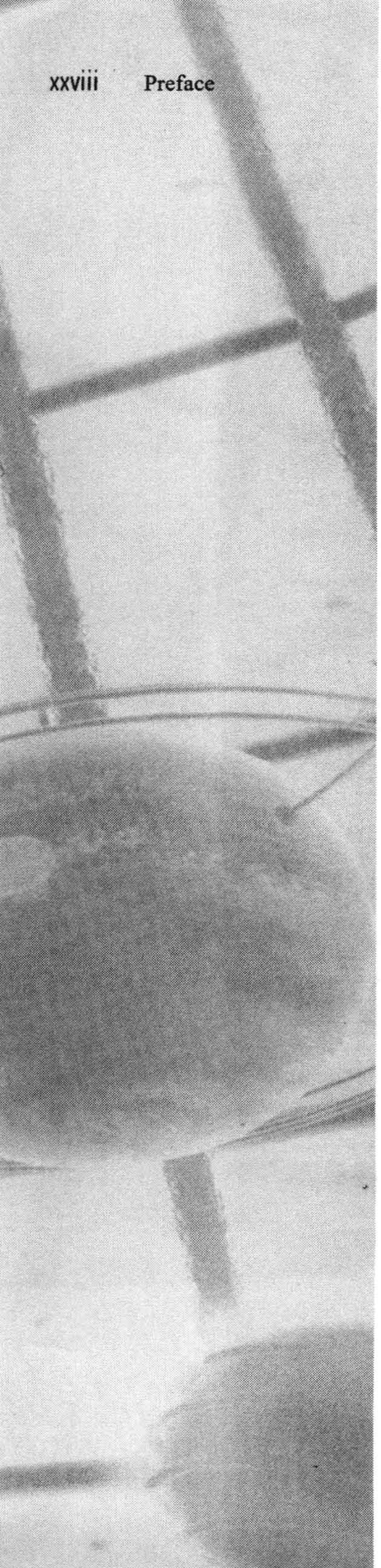
Chapter 8, formerly Chapter 7, is largely unchanged.

The new Chapter 9, formerly Chapter 8, contains an extensive treatment of scene graphs and uses a scene graph developed by one of my students, Ye Cong, as a class project. I debated about using a "standard" scene graph, such as those in VRML or Java3D, but came to the conclusion that there is no standard and it is unclear whether or not these candidates will survive as long as the third edition. However, I do believe the concepts are important and thus decided to go with a simple scene graph that students could work with using code I could place on the Web.

Chapter 10 now contains an introduction to subdivision curves and surfaces. Chapters 11 and 12 are largely unchanged from the second edition.

Chapter 13 is new. It contains expanded introductions to ray tracing and radiosity, including a discussion of the rendering equation, which was covered in





Chapter 6 in the previous two editions. I have added an introduction to image-based rendering, an area of great recent interest that encompasses a multitude of approaches. I have added a section on parallel rendering that is related to my own interests and also should gain importance as clusters of PCs with commodity graphics cards replace supercomputers built with specialized hardware.

Appendix A contains added examples. Appendix B now contains a short section on eigenvalues and eigenvectors to support topics such as smooth rotation.

Supplements

The CD that accompanies the book includes software tools (recent distributions of GLUT and MESA), source code and Windows binaries for the examples in the text (and some additional examples), OpenGL man pages (in html and ps formats), my OpenGL tutorial (in pdf format) from SIGGRAPH 2001, and Nate Robins' OpenGL tutorials.

The following support material is available to all readers of this book.

- Program code
- Solutions to selected exercises

Please visit http://www.aw.com/cssupport for more information on obtaining these supplements.

Additional support materials are only available to instructors adopting this textbook for classroom use. Please contact your school's Addison-Wesley representative for information on obtaining access to this material, including

- Solutions to all exercises
- Figure files in three formats so that you can create your own lecture notes
- Tutorial lecture notes on OpenGL.

All of this information and more is available on my Web site

http://www.cs.unm.edu/~angel

Programs from the book, as well as additional sample programs, are available over the Internet via anonymous ftp at:

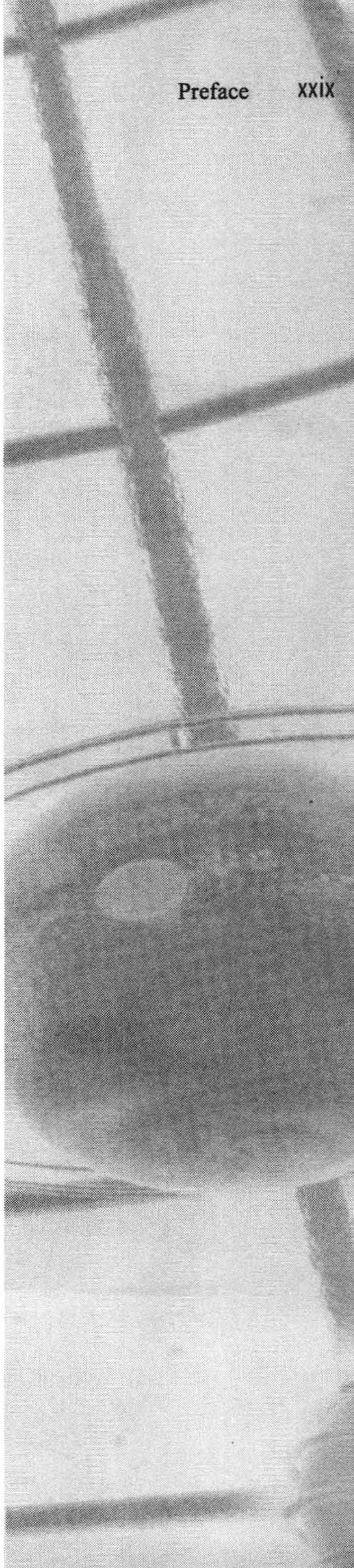
ftp.cs.unm.edu under pub/angel/BOOK

I welcome suggestions regarding other supplements that readers would find useful, as well as comments on the book itself:

angel@cs.unm.edu

Acknowledgments

I have been fortunate over the past few years to have worked with wonderful students at the University of New Mexico. They were the first to get me interested in OpenGL, and I have learned much from them. They include Hue



Bumgarner-Kirby (Walker), Ye Cong, Pat Crossno, Tommie Daniel, Lisa Desjarlais, Kim Edlund, Lee Ann Fisk, Maria Gallegos, Brian Jones, Christopher Jordan, Max Hazelrigg, Sheryl Hurley, Thomas Keller, Ge Li, Pat McCormick, Al McPherson, Martin Muller, David Munich, Jim Pinkerton, Jim Prewett, Dave Rogers, Hal Smyer, Dave Vick, and Brian Wylie. Many of the examples in the Color Plates section were created by these students.

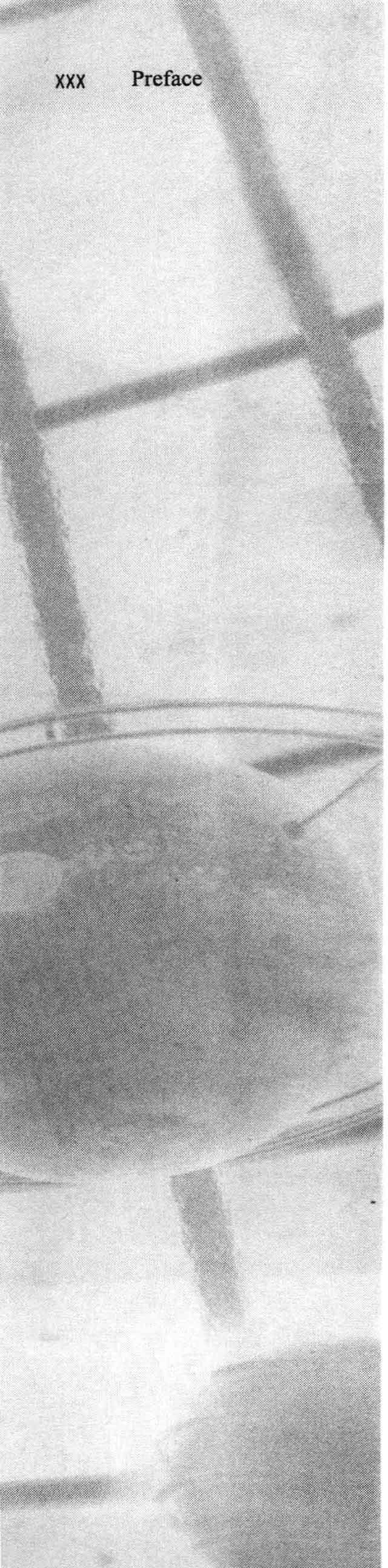
The first edition of this book was written during my sabbatical; various parts were written in five different countries. The experience speaks well for the versatility of portable computers and the universality of the Internet. Nevertheless, the task would not have been accomplished without the help of a number of people and institutions that made their facilities available to me. I am greatly indebted to Jonas Montilva and Chris Birkbeck of the Universidad de los Andes (Venezuela), to Rodrigo Gallegos and Aristides Novoa of the Universidad Tecnologica Equinoccial (Ecuador), to Long Wen Chang of the National Tsing Hua University (Taiwan), and to Kin Hong Wong and Pheng Ann Heng of the Chinese University of Hong Kong. Ramiro Jordan of ISTEC and the University of New Mexico made possible many of these visits. John Brayer and Jason Stewart at the University of New Mexico and Helen Goldstein at Addison-Wesley somehow managed to get a variety of items to me wherever I happened to be. My Web site contains a description of my adventures writing the first edition.

The second and third editions was written in New Mexico and benefited greatly from the feedback of hundreds of instructors and students who used these editions and sent me their comments. Many of the changes from earlier editions are responses to their requests and comments.

Silicon Graphics and Apple Computer were generous in making equipment available to me. Conix Enterprises, Portable Graphics, Template Graphics, and Metrowerks were kind enough to provide software so that I could test sample programs on a variety of platforms. David Kirk and Mark Kilgard at NVidia were kind enough to provide cards for testing many of the recent algorithms.

A number of other people provided significant help. I thank Ben Bederson, Gonzalo Cartagenova, Tom Caudell, Kathi Collins, Kathleen Danielson, Roger Ehrich, Robert Geist, Chuck Hansen, Mark Henne, Bernard Moret, Dick Nordhaus, Helena Saona, Dave Shreiner, Vicki Shreiner, Gwen Sylvan, and Mason Woo. Mark Kilgard, Brian Paul, and Nate Robins are owed a great debt by the OpenGL community for creating software that enables OpenGL code to be developed over a variety of platforms.

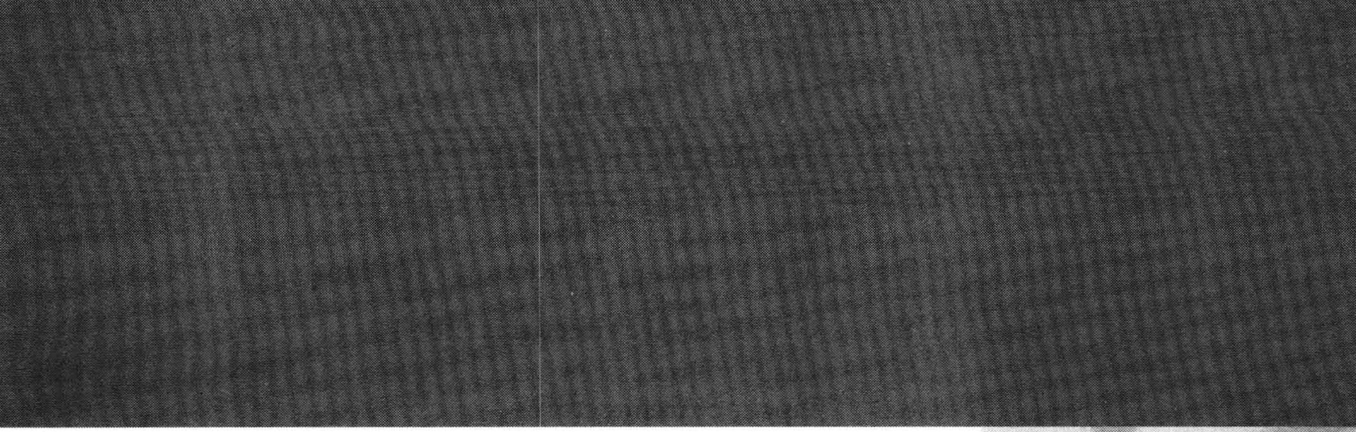
At the University of New Mexico, the High Performance Computing Center and IBM provided support for many of my visualization projects. The Computer Science Department, the National Science Foundation, Sandia National Laboratories, and Los Alamos National Laboratory have supported many of my students and research projects that led to parts of this book. David Beining and the Lodestar Planetarium have provided tremendous support for the Full Dome Project (Sheryl Hurley, Christopher Jordan, Laurel Ladwig, Hue (Bumgarner-Kirby) Walker) that provided the cover images and some of the images in the Color Plate section.

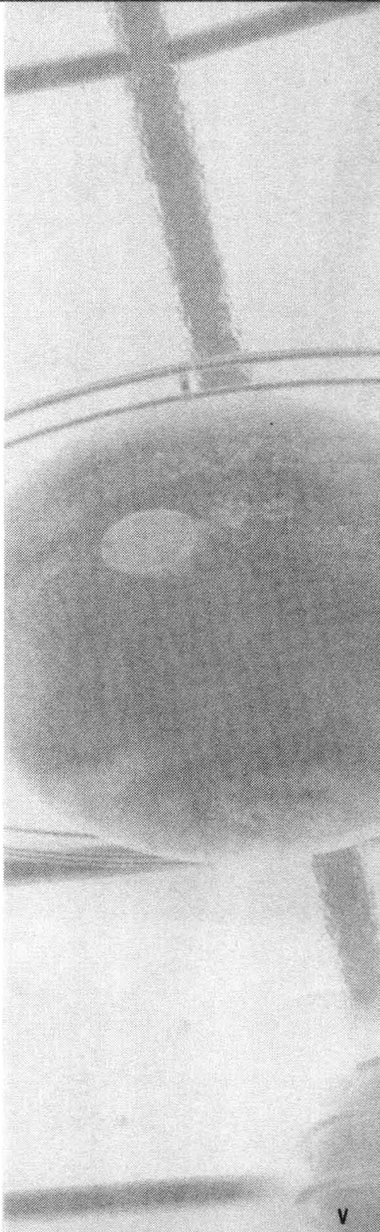


Reviewers of my manuscript drafts provided a variety of viewpoints on what I should include and what level of presentation I should use. These reviewers included Hamid Arabnia (University of Georgia), Wayne Carlson (Ohio State University), Jim X. Chen (George Mason University), Norman Chin (Silicon Graphics), Andrew Duchowski (Clemson University), Matthew Evett (Eastern Michigan University), Michael Garland (University of Illinois), Scott Grissom (University of Illinois, Springfield), Ardeshir Goshtasby (Wright State University), John C. Kelly, Jr. (North Carolina A&T State University), Kwan-Liu Ma (University of California, Davis), David McAllister (North Carolina State University), Tom McReynolds (Silicon Graphics), Jim Miller (University of Kansas), Dick Phillips (formerly Los Alamos National Laboratories), Yang Wang (Southwest Missouri State University), Jane Wilhelms (University of California, Santa Cruz), Edward Wong (Brooklyn Polytechnic Institute), and Mike Wozny (Rensselaer Polytechnic Institute). Although the final decisions may not reflect their views—which often differed considerably from one another—each reviewer forced me to reflect on every page of the manuscript.

I acknowledge the entire production team at Addison-Wesley; these people, in addition to doing their usual great job, took on the additional burden of working with someone who was never in one place for long. My editors, Peter Gordon and Maite Suarez-Rivas, have been a pleasure to work with through three editions of this book and the Open Primer. Through three books, Paul Anagnostopoulos at Windfall Software has always been more than helpful in assisting me with TeX problems. I am especially grateful to Lyn Dupré. I am not a natural writer. If the readers could see the original draft of this book, they would understand the wonders that Lyn does with a manuscript.

My wife, Rose Mary Molnar, created the figures for my first graphics book, many of which form the basis for the figures in this book. Wisely choosing not to fight over use of our only notebook computer, she was able to preserve our relationship and to contribute in a thousand other ways.





CONTENTS