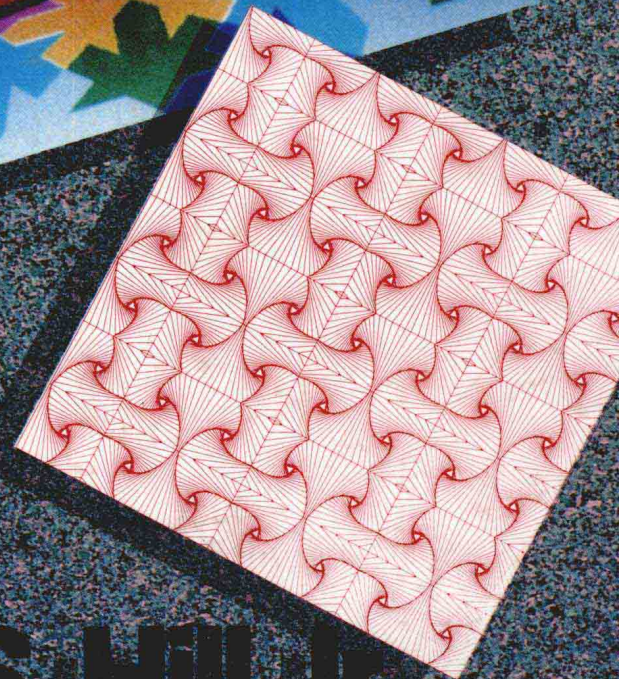
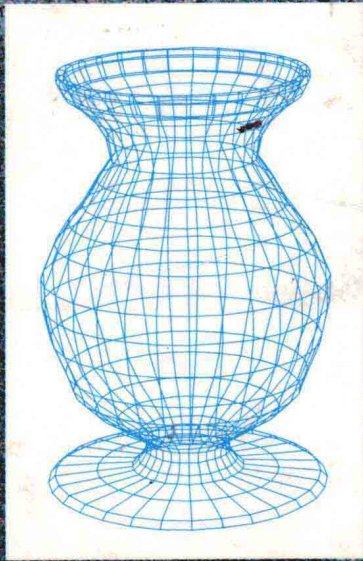
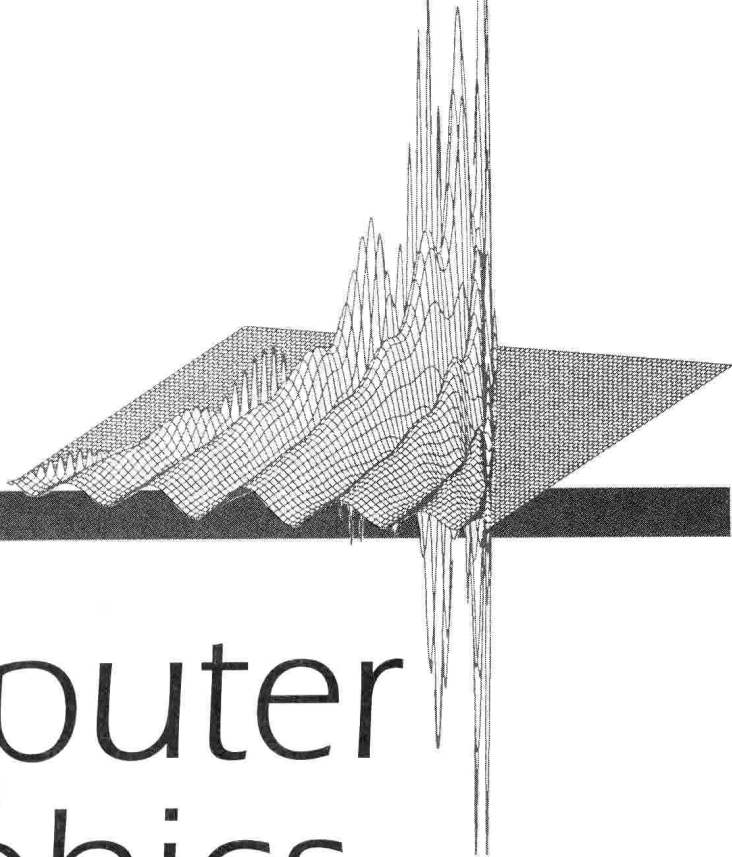


# COMPUTER GRAPHICS



F.S. WILKINS JR.



# Computer Graphics

Francis S. Hill, Jr.

Department of Electrical and Computer Engineering

Macmillan Publishing Company  
New York

Collier Macmillan Publishers  
London

Editor: John Griffin  
Production Supervisor: Ron Harris  
Production Manager: Pam Kennedy Oborski  
Text Designer: Jack Ehn  
Cover Designer: Sheree L. Goodman

This book was set in 10/12 Melior by T.S.I. Graphics and printed and bound by Von Hoffmann Press. The cover was printed by Von Hoffmann Press.

Copyright © 1990, Macmillan Publishing Company, a division of Macmillan, Inc.

Printed in the United States of America

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher.

Macmillan Publishing Company  
866 Third Avenue, New York, New York 10022

Collier Macmillan Canada, Inc.

Library of Congress Cataloging-in-Publication Data  
Hill, Francis S.

Computer graphics / Francis S. Hill, Jr.

p. cm.

Bibliography: p.

Includes index.

ISBN 0-02-354860-6

1. Computer graphics. I. Title.

T385.H549 1990

89-31610

006.6—dc20

CIP

Printing: 1 2 3 4 5 6 7 8

Year: 0 1 2 3 4 5 6 7 8 9

TO  
Merilee  
and to  
Greta, Jessie, and Rosy

# Preface

Computer graphics is a fascinating area of computer science. It is widely used as a tool for visualizing information in a broad variety of fields, including science and engineering, medicine, architecture, and entertainment. Interactive graphics programs let people work with computers in a natural manner: a user can supply information to the program through simple hand movements and receive information back through pictures. Computer graphics is helping to change the way people perceive and use computers, and it is available on even the lowest-cost computers today, making it readily accessible to everyone.

This book teaches the concepts and techniques of computer graphics. It teaches people how to write programs that produce graphical pictures and images of many kinds of information. It is designed as a text for either a one- or two-semester course in computer graphics at the junior, senior, or graduate level. It can also be used for self-study. It is aimed principally for students majoring in computer science or engineering but will also suit students in other fields, such as physics, mathematics, business, and art.

## Prerequisites

In general, the reader should have at least one semester of experience writing computer programs in a modern language such as Pascal, C, or FORTRAN. Some experience with elementary data structures such as linked lists is desirable but not essential, since alternative approaches based on arrays are described.

The reader should have the equivalent of one year of college mathematics; knowledge of elementary algebra, geometry, and trigonometry is assumed. Some exposure to calculus and matrices is useful but not necessary, as an appendix introduces those parts of matrix theory that are used in the book.

## Philosophy

The basic philosophy of this book is that graphics programming is learned by doing it: One must write and test real programs that drive real graphics devices to comprehend fully what is going on. One of the principal goals of the book is to show readers how to translate geometric concepts first into a clear mathematical expression and then into program code that works on a computer. Readers first learn how to develop simple routines to produce pictures on whatever graphics display devices they have available. Then methods for producing drawings of ever more complex objects and data are presented in a step-by-step fashion.

## Code Fragments

Over 140 examples of tested code fragments, written in Pascal, serve as examples to show one way to implement an algorithm. These fragments appear throughout the text and illustrate real-life applications of the theory.

## Exercises and Problems

More than 440 drill exercises appear throughout the book. These are of the “stop-and-think” variety that don’t require programming and allow readers to

self-test their grasp of the material. Over 180 programming problems appear at the end of chapters. These are suitable for homework assignments and range from the simple to the challenging.

## Illustrations

Numerous two- and four-color drawings, photographs, and stereo drawings are used throughout the text to illustrate technique and theory. Computer graphics is of course a very visual discipline, and the figures help to fine-tune the reader's ability to visualize geometric relationships. This is particularly true for three-dimensional graphics, where the stereo pictures help enormously to reveal the geometric concepts being discussed. For suggestions on how to view the stereo pictures, please see the note that precedes the table of contents.

## Device Independence

A college graphics laboratory may contain a sizable number of graphics devices, but only a few of each type or model. It can be frustrating for the instructor and student alike to write different programs for so many idiosyncratic devices. To reduce such frustration levels this book takes a device-independent approach. All the graphics in the first part of the book are built out of applications of a single short routine called *lineNDC()* that draws a line between two points. All programming of applications accesses the graphics capability of a device through this routine. A different version of this routine—called a device driver—is fashioned for each type of device. The student can write programs without having to know beforehand which devices will be available. To test a program the student finds the nearest available graphics device, activates the appropriate driver, and is ready to run the application.

Appendix 1 gives examples of actual simple device drivers for several classes of graphics output devices. (Additional ones are provided in the accompanying instructor's manual.) This information is useful in helping the reader or instructor over the first hurdle of producing simple pictures on whatever devices are available.

## Complete, Up-to-Date Organization

Computer graphics equipment changes rapidly: Each year we see ever more dazzling displays and ever faster computations. But while pictures of equipment and the images they produce slip quickly out of date, the underlying theory and mathematics of computer graphics has become very stable. The body of knowledge required to produce effective graphics programs rests on a small set of key mathematical topics, including vectors, parametric representations, affine transformations, homogeneous coordinates, and projections. Great care has been given in this book to both clarifying and exposing the beauty of these topics.

## Flexible Organization

There is much more in this book than can be covered in a one-semester course. The book has been arranged so the instructor can select different groups of chapters for close study, depending on the interests and backgrounds of a class.

Several such paths through the book are suggested here.

- For a one-semester undergraduate course: Chapters 1, 2, 3, and 4 and parts of 7, 10, 11, and 12.
- For a two-semester undergraduate course add the rest of Chapters 7, 9, 15, 16, 17, and 18.
- For a one-semester graduate course: Chapters 1, 2, 3, 4, 6, 7, 9, 10, 11, and 12.
- For a two-semester graduate course add Chapters 8, 13, 15, 16, 17, and 18.
- For those interested in emphasizing modeling and 3D graphics, this material is concentrated in Chapters 10, 12, 15, 17, and 18.
- For emphasis on raster graphics, Chapters 13, 15, 16, 17, and 18 are of particular interest.

All suggested paths include Chapters 1, 2, and 3 as fundamental. Chapter 1 can be read quickly, with the instructor focusing early on notions of NDC, device drivers, windows and viewports, and producing simple line drawings. The crucial parts of Chapter 4 are dealing with polygons and the important technique of representing a curve parametrically. The material of Chapter 5 is not essential to the development of later ideas, although many students—particularly those with an interest in the arts—vastly enjoy producing such pictures. Chapter 6 is important for gaining an understanding of interactive graphics, but if an instructor prefers to focus only on the production of pictures this chapter can be omitted.

Chapter 7 contains essential information on the use of vectors in graphics. Even students already versed in vectors should peruse it to see how valuable ordinary vector tools are in graphics algorithms. Chapter 8 makes a good project for students interested primarily in two-dimensional graphics. Chapter 9 is important for an understanding of smoothly varying surfaces but may be omitted if interest is greatest in viewing faceted models. Chapter 10 introduces the fascinating world of three-dimensional (3D) graphics based on wireframe models.

Chapter 11 discusses in detail another pivotal tool in graphics, the affine transformation, and shows how it is applied in graphics applications. And Chapter 12 moves deeper into 3D graphics, introducing the important “synthetic camera,” and developing the various types of projections used in graphics.

In Chapter 13 we change gears and discuss a variety of topics peculiar to raster graphics, including scan conversion, region filling, antialiasing, and the powerful BitBlt operation. Chapter 14 discusses curve and surface design based on Bezier and B-spline methods.

Chapter 15 shows how solid objects that are modeled as “polygon meshes” can be drawn with smooth shading to achieve greater realism. The classical lighting models, which involve diffuse and specular components of light, are introduced, and Gouraud and Phong shading are described.

Methods for defining and classifying colors numerically are discussed in Chapter 16, leading to the CIE standard chromaticity diagram and its various uses. Chapter 17 attacks the surface elimination problem and presents several specific algorithms in detail.

In Chapter 18 we develop a complete working ray tracer, for achieving stunning visual realism. The development is incremental, so the reader can build it in small working pieces. The chapter also shows how to include the effects of mirrorlike and transparent objects.

## Supplements

An accompanying instructor's manual provides solutions to most exercises and suggests additional projects. Complete demonstration programs of techniques developed in the text are explained and listed. Reprints of several articles on interesting geometric ideas provide the instructor with engaging background material for the student. Also included in the manual are guides to additional device drivers, to help the instructor in developing drivers for a wider variety of graphics device.

A diskette containing many of the code fragments and demonstration programs accompanies the instructor's manual. The demonstration programs are suitable for use in the classroom, and students can work with them and enhance them as well.

## Acknowledgments

This book has grown out of notes used in a course I have been teaching at the University of Massachusetts for the last nine years. During this time a large number of students have helped to develop demonstrations and make suggestions for improving the courses. They have also produced many exquisite graphical samples, some of which appear here. Some students who have been particularly helpful are Tarik Abou-Raya, Earl Billingsley, Dennis Chen, Scott Davidson, Daniel Dee, Brett Diamond, Bruce Filgate, Jay Greco, Marc Infield, Tom Kopec, Adam Lavine, Tuan Le, Andreas Meyer, John Michael, David Mount, Bruce Nichol, Mike Purpura, Chris Russell, Russell Turner, Bill Verts, Shel Walker, Ken Ward, and Clay Yost.

I apologize for any inadvertent omissions.

Several colleagues have provided inspiration and guidance during the germination of the book. I am particularly grateful to Charles Hutchinson for his support in starting the graphics effort at the university, to Michael Wozny for his enthusiasm and encouragement in its development, and to Charlé Rupp for the many creative ideas in graphics he passed on to me. I would especially like to thank Daniel Bergeron, Robert Wilke, Tim Clement, and Brian Barsky, who made substantial contributions to the coherence and readability of the book. I would like to thank the following individuals, and many others who are not mentioned by name, for their advice and help: Marc P. Armstrong, University of Iowa; J. Eugene Ball, University of Delaware; Brian A. Barsky, University of California–Berkeley; Marc Berger, University of Colorado, R. Daniel Bergeron, University of New Hampshire; Tim Clement, University of Manchester; John T. Demel, The Ohio State University; Edward N. Ferguson, University of Maine; Georges Grinstein, University of Lowell; Mark W. Koch, Clarkson University; Leo J. LaFrance, New Mexico State University; Peichung F. Lai, University of Alabama; Ralph E. Lee, University of Missouri–Rolla; Joel Neisen, University of Minnesota; Chuck Nelson, University of Idaho; Spencer W. Thomas, University of Michigan; Deborah Walters, State University of New York–Buffalo; Frank G. Walters, University of Missouri–Rolla; Robert Wilke, The Ohio State University.



Special thanks to my editor, John Griffin, for his continuing guidance and encouragement during the preparation of the book, and to Ron Harris, whose expertise and care during production have markedly improved it. Anne Dolan-Niles helped enormously organizing the voluminous correspondence connected with the book, and Andy Casiello produced many of the photographs.

Finally, thanks to my parents, to my wife Merilee, and to our children Greta, Jessie, and Rosy, for all their patience and support while this book slowly took shape.

F. S. H., Jr.

## **Note to the reader**

### How to View the Stereo Pictures

It can be challenging to visualize things in three-dimensional (3D) space, and figures in a book are limited to showing projections of objects on a flat page. One way to counter this limitation dramatically is to view images stereoscopically. The eye–brain system is remarkable: If slightly different views of a scene are shown to the left and right eyes, a scene can appear to “jump out of the page” with a convincing sense of depth.

Many stereoscopic figures appear throughout the book to clarify discussions of 3D situations. They look like a pair of almost identical figures placed side by side. To gain the full value of these stereo pictures allow (coerce) your left eye to look at the left-hand one and your right eye to look at the right-hand one. This may take some practice, since we go about our lives focusing both eyes on the same points. Some people catch on quickly; others, after many bleary-eyed attempts; some people, never. Of course the figures still help to clarify the discussion even without the stereo effect.

One way to practice is to hold the index fingers of each hand upright in front of you, about 2 inches apart, and to stare “through them” at a blank wall in the distance. Each eye sees two fingers, of course, but two of the fingers seem to overlap in the middle. This overlap is precisely what is desired when looking at stereo figures: Each eye sees two figures, but the middle ones are brought into perfect overlap. When the middle ones fuse together like this, the brain constructs out of them a single 3D image. Some people find it helpful to place a piece of white cardboard between the two figures and to rest their nose on it. The cardboard barrier prevents each eye from seeing the image intended for the other eye.

An inexpensive cardboard viewer is available from the Taylor–Merchant Corporation; 212 West 35 Street; New York, NY 10001. The instructor can request that the college bookstore stock these viewers.

## Brief Contents

1	Introduction to Computer Graphics	1
2	Graphics Devices and Their Control	23
3	Getting Started Making Pictures	60
4	Building and Drawing Curves	103
5	Approaches to Infinity	141
6	On Interaction in Graphics	174
7	Vectors and Their Use in Graphics	213
8	Two-Dimensional Ray Tracing: Reflections in a Chamber	267
9	Modeling Surfaces	279
10	Three-Dimensional Graphics	305
11	Transformations of Pictures	327
12	Three-Dimensional Viewing with the Synthetic Camera	384
13	Raster Graphics Techniques	425
14	Curve and Surface Design for CAD	482
15	Adding Faces for Visual Realism	526
16	Color Theory: Coloring Faces	563
17	Hidden Surface Elimination	586
18	Introduction to Ray Tracing	615

## Appendixes

1	Examples of Elementary Device Drivers	671
2	Review of Matrix Algebra	681
3	Summary of Pascal Data Types	689
4	Example Solids	699
5	On Spherical Coordinates	708
6	Advanced Clipping Algorithms	711
7	A Closer Look at Graphics Standards	723
8	The Platonic Solids	732

# Contents

1	Introduction to Computer Graphics	1
1.1	What Is Computer Graphics?	2
1.1.1	The Major Ingredients	2
1.1.2	The Graphics Workstation	8
1.2	Examples of Graphics Applications	9
1.3	Overview of Graphics Software	16
2	Graphics Devices and Their Control	23
2.1	Introduction	24
2.2	Graphics Devices	24
2.2.1	Line-Drawing Displays	24
2.2.2	Raster-Scan Displays	28
2.2.3	Controlling the Frame Buffer	34
2.3	Developing Device Drivers	36
2.4	Graphics Output Primitives and Attributes	43
2.4.1	The GKS Output Primitives	43
2.5	Graphics Input Devices	52
2.5.1	Types of Logical Input Functions	52
2.5.2	Types of Physical Input Devices	55
2.6	Summary	58
3	Getting Started Making Pictures	60
3.1	Introduction	61
3.2	Simple Line Drawing	61
3.3	The World Coordinate System	64
3.3.1	Windows and Viewports	66
3.3.2	Mapping from a Window to a Viewport	70
3.4	Clipping a Line	72
3.4.1	The Cohen–Sutherland Clipping Algorithm	73
3.4.2	A Bonus—The Golden Ratio	78
3.5	Adding More Drawing Tools	80
3.5.1	The Current Position in World Coordinates	80
3.5.2	Defining <i>Polyline()</i>	81
3.5.3	Relative Moves and Draws	82
3.6	Relative Polar Coordinates: Turtlegraphics	84
3.6.1	An Application of Turtlegraphics: Polyspirals	89

3.7	Line and Bar Charts	90
	3.7.1 Line Graphs	93
	3.7.2 Bar Graphs	96
3.8	Summary	97
4	Building and Drawing Curves	103
4.1	Introduction	104
4.2	Polygons and Their Offspring	104
	4.2.1 Building and Drawing $n$ -gons	105
	4.2.2 Rosettes	107
4.3	Drawing Figures Based on Circles	110
	4.3.1 An Electrical Engineering Application of <i>Circle()</i>	111
	4.3.2 Drawing Pie Charts	112
4.4	Parametric Representations of Curves	115
	4.4.1 Lines, Rays, and Line Segments	116
	4.4.2 Circles and Ellipses	118
	4.4.3 Drawing Curves Represented Parametrically	121
	4.4.4 The Exquisite Superellipse	122
	4.4.5 Polar Coordinate Shapes	124
	4.4.6 Circles Rolling Around Circles	128
	4.4.7 Playful Variations on a Theme	130
4.5	Testing for Intersections of Line Segments	131
	4.5.1 The Problem	131
4.6	Summary	135
5	Approaches to Infinity	141
5.1	Introduction	142
5.2	Tiling the Plane	142
	5.2.1 Drawing Simple Tessellations	144
	5.2.2 More General Tilings	145
5.3	Recursively Defined Curves	149
	5.3.1 Koch Curves	149
	5.3.2 C-Curves and Dragons	152
	5.3.3 Space-Filling Curves	153
	5.3.4 Reptiles	157
5.4	Fractals	159
	5.4.1 Self-similarity in Curves	159
	5.4.2 Fractal Trees	165
5.5	The Mandelbrot Set	167
5.6	Summary	170
6	On Interaction in Graphics	174
6.1	Introduction	175
6.2	Introduction to the <i>Locate</i> function	175
	6.2.1 Using <i>Locate()</i> for Instancing	177
	6.2.2 Application: Interactively Creating Polylines	178

6.2.3	The <i>Stroke</i> Input Function	179
6.2.4	Coordinate Systems Used for <i>Locate()</i> and <i>Stroke()</i>	180
6.3	The Graphics Cursor: Prompts and Echoes	182
6.3.1	Initial Position of the Cursor	182
6.3.2	Cursor Shapes	183
6.4	A Simple <i>Locator</i> Driver	184
6.5	A Glimpse at a Mouse Driver	186
6.6	Using <i>Locate</i> to Select an Object	187
6.6.1	Geometric Picking Using a Distance Criterion	188
6.6.2	Programming Project: A Polyline Editor	189
6.6.3	The Functional Specification of <i>Politor</i>	190
6.7	Geometric Picking Using Containment	193
6.8	Menu Design and Selection in Graphics Applications	193
6.8.1	Screen Layout and Menu Design	195
6.8.2	Menu Selection Using <i>LocateNDC()</i>	198
6.9	A CAD Application: Creating Networks Graphically	199
6.9.1	Placement of the Elements: Pick and Place	201
6.10	Working with Pictures As Segments	204
6.10.1	What Is a Segment?	204
6.10.2	Automatic Redisplay of Segments	205
6.10.3	Segment Attributes	206
6.11	The <i>Pick</i> Input Function	208
6.12	Summary	210
7	Vectors and Their Use in Graphics	213
7.1	Introduction: Vectors and Their Manipulation	214
7.2	Operations with Vectors	216
7.2.1	Adding Two Vectors	217
7.2.2	Scaling a Vector	218
7.2.3	Subtracting Vectors	218
7.2.4	The Magnitude (Length) of a Vector	218
7.2.5	Linear Combinations of Vectors	220
7.2.6	Vectors Versus Points	223
7.2.7	Convex Sets and Convex Hulls	229
7.3	The Dot Product of Two Vectors	231
7.3.1	Properties of the Dot Product	232
7.4	Applications of the Dot Product	233
7.4.1	The Angle Between Two Vectors	233
7.4.2	The Sign of $\mathbf{a} \cdot \mathbf{b}$ , and Perpendicularity	234
7.4.3	Projecting and Resolving Vectors	235
7.4.4	Application of Projection: Reflections	237
7.4.5	The Point Normal Form for a Line and a Plane	238
7.4.6	Inside–Outside Half-space Test for a Point	243
7.4.7	Application: Line Clipping to Convex Windows	245

7.5	Using Three-Dimensional Vectors	249
7.5.1	Left-Handed Versus Right-Handed Coordinate Systems	250
7.6	The Cross Product of Two Vectors	251
7.6.1	Geometric Interpretation of the Cross Product	252
7.6.2	The Scalar Triple Product	253
7.7	Applying the Cross Product and Scalar Triple Product	254
7.7.1	Finding the Normal to a Plane	255
7.7.2	The Simplest Polygons: Triangles	256
7.8	Applications to Polygons	256
7.8.1	Testing the Planarity of a Polygon	257
7.8.2	Find the Normal to a Polygonal Face	258
7.8.3	Detecting a Left Versus a Right Turn	259
7.8.4	Test the Convexity of a Polygon	260
7.8.5	Computing the Area of a Polygon	261
7.9	Summary	264
8	Two-Dimensional Ray Tracing: Reflections in a Chamber	267
8.1	A Ray-Tracing Experiment	268
8.1.1	The Intersection of a Ray with a Line	270
8.1.2	The Intersection of a Ray with a Circle	273
8.1.3	Building the Ray-Tracing Experiment	275
8.2	Variation on a Theme: Elliptipool	276
9	Modeling Surfaces	279
9.1	Introduction	280
9.2	Parametric Representations for Surfaces	280
9.3	The Parametric Form for a Plane	282
9.3.1	Planar Patches	284
9.4	Introduction to Curved Surfaces	285
9.5	Ruled Surfaces	286
9.5.1	Cylinders	287
9.5.2	Cones	289
9.5.3	Bilinear Patches	290
9.6	The Normal Vector to a Surface	291
9.7	Surfaces of Revolution	294
9.8	The Quadric Surfaces	296
9.8.1	Normal Vectors to Quadric Surfaces	298
9.8.2	Some Notes on the Quadric Surfaces	299
9.9	The Superquadrics	300
9.9.1	Normal Vectors for the Superquadrics	302

9.10	Mathematical Functions of Two Variables	303
9.11	Summary	304
10	Three-Dimensional Graphics	305
10.1	Introduction	306
10.2	Wireframe Models	307
10.3	Drawing Wireframes: Introduction to Projections	308
10.3.1	The Simplest Orthographic Projections	309
10.3.2	Simple Perspective Projections	311
10.4	Further Wireframe Modeling of Objects	315
10.4.1	Modeling a Prism	315
10.4.2	The Platonic Solids	316
10.5	Drawing Surfaces	317
10.5.1	Drawing Surface Patches	317
10.5.2	Drawing Curved Surfaces Using Contours	320
10.6	Summary	324
11	Transformations of Pictures	327
11.1	Introduction	328
11.2	Two-Dimensional Transformations	328
11.3	The Affine Transformations	330
11.3.1	Translation	331
11.3.2	Scaling	332
11.3.3	Rotation	333
11.3.4	Shearing	335
11.3.5	The Inverse of an Affine Transformation	336
11.4	Some Useful Properties of Affine Transformations	338
11.5	Creating and Applying Affine Transformations	342
11.5.1	Transforming Polylines	343
11.6	Composing Affine Transformations	344
11.6.1	Decomposing Affine Transformations	350
11.7	A Valuable Unification: Homogeneous Coordinates	352
11.7.1	The Underlying Notion of Homogeneous Coordinates	355
11.8	Application: Transforming Hierarchical Objects	357
11.9	Three-Dimensional Affine Transformations	363
11.9.1	Properties of Three-Dimensional Affine Transformations	363
11.9.2	Translation	364
11.9.3	Scaling	364
11.9.4	Three-Dimensional Shears	364
11.9.5	Rotation	365
11.9.6	Building and Using Three-Dimensional Affine Transformations	368
11.9.7	Using Rotations for Better Views of Three-Dimensional Objects	371
11.10	Unified Composition of Three-Dimensional Affine Transformations	374
11.11	Some Interesting Nonaffine Transformations	376
11.11.1	Fish-Eye Transformation	376



11.11.2	Inversion in a Unit Circle	377
11.11.3	False Perspective	379
11.12	Summary	380
12	Three-Dimensional Viewing with the Synthetic Camera	384
12.1	Introduction	385
12.2	The Synthetic Camera Approach	385
12.2.1	Quantitative Description of the Synthetic Camera	386
12.3	How Does a User Specify the Camera?	388
12.3.1	An Alternative: Using Angles to Specify Direction	389
12.4	Describing Object Points in Viewing Coordinates	391
12.4.1	Representing Objects in Viewing Coordinates	393
12.5	Viewing the Object	394
12.6	The Geometry of Perspective Projections	395
12.6.1	Eye on the $N$ -axis	396
12.6.2	Rays Through the Eye, and Pseudodepth	397
12.6.3	Parallel Projections	403
12.7	Taxonomy of Projections	405
12.7.1	One-, Two-, and Three-Point Perspective	405
12.8	More on Parallel Projections	410
12.8.1	Orthographic Projections	410
12.8.2	Axonometric Views	411
12.8.3	Oblique Projections	413
12.9	Clipping Lines to the View Volume	414
12.9.1	The View Volume	414
12.9.2	Summary: What Have We Accomplished So Far?	416
12.9.3	The Normalization Transformation	417
12.9.4	The Clipping Process	419
12.10	Summary	422
13	Raster Graphics Techniques	425
13.1	Introduction	426
13.2	Introduction to Scan Conversion	427
13.2.1	Line-Drawing Algorithms	428
13.2.2	Bresenham's Line Algorithm	428
13.2.3	Scan Converting Circles and Ellipses	433
13.3	Methods for Filling Regions	436
13.3.1	Defining Regions	436
13.3.2	A Recursive Flood-Fill Algorithm	437
13.3.3	Using Coherence: Region Filling Based on Runs of Pixels	439
13.3.4	Filling Polygon-Defined Regions	441
13.3.5	Filling Several Polygons at Once	447